

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

\_\_\_\_\_  
(повна назва інституту/факультету)

\_\_\_\_\_  
(повна назва кафедри)

«На правах рукопису»  
УДК \_\_\_\_\_

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(ініціали, прізвище)

“ ” \_\_\_\_\_ 20\_\_

р.

## Магістерська дисертація

зі спеціальності (спеціалізації) \_\_\_\_\_  
(код і назва спеціальності)

на тему: Автоматизована система контролю розрахунків для мережі магази-  
нинів роздрібно́ї торгівлі

Виконав: студент 6 курсу, групи ІА-82мп  
(шифр групи)

Шихутського Сергія Олександровича

\_\_\_\_\_  
(прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

Науковий керівник к.т.н., доц. Новацький А.О.  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисер-  
тації немає запозичень з праць інших авторів  
без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2019 року

Національний технічний університет України  
«Київський політехнічний інститут  
імені Ігоря Сікорського»

Факультет (інститут) \_\_\_\_\_  
(повна назва)

Кафедра \_\_\_\_\_  
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною (освітньо-науковою) програмою

Спеціальність (спеціалізація) \_\_\_\_\_  
(код і назва)

ЗАТВЕРДЖУЮ  
Завідувач кафедри

\_\_\_\_\_  
(підпис) (ініціали, прізвище)

«\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студенту**

\_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема дисертації Автоматизована система контролю розрахунків для мережі магазинів роздрібної торгівлі

науковий керівник дисертації \_\_\_\_\_ ,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «\_\_» \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Строк подання студентом дисертації \_\_\_\_\_

3. Об'єкт дослідження побудова розподілених програмних продуктів, базуючись на існуючій архітектурі

4. Предмет дослідження (вихідні дані для магістерської дисертації за освітньо-професійною програмою) комплекс розподілених систем, що знаходяться в загальній екосистемі, та проектування системи в рамках обраної екосистеми.

5. Перелік завдань, які потрібно розробити огляд предметної області та формулювання мети дослідження, ознайомлення з методом проектування розподілених систем та інструментарієм, проектування архітектури системи, вибір технологій розробки системи, аналіз можливості впровадження результатів дослідження у вигляді стартап-проекту

6. Орієнтовний перелік ілюстративного (графічного) матеріалу 29 рисунки, 18 таблиць, 8 додатків

7. Орієнтовний перелік публікацій \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

8. Дата видачі завдання 05.09.2019

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка

Студент

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(ініціали, прізвище)

Науковий керівник дисертації

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(ініціали, прізвище)

## РЕФЕРАТ

Дипломна робота освітньо-кваліфікаційного рівня «?» на тему «Автоматизована система контролю розрахунків для мережі магазинів роздрібної торгівлі». Робота містить в собі ? сторінок звіту, 11 розділів, 29 ілюстрацій, 18 таблиць, та 8 додатків.

Мета роботи – розробка технології самообслуговування покупців в мережі магазинів роздрібної торгівлі, яка б ефективно економила б час споживача, приносила прибуток самому магазину, а також відкривала нові можливості до таргетованої взаємодії зі споживачем.

Предмет дослідження – способи побудови розподілених програмних продуктів, що базуються на існуючій архітектурі.

Об’єкт дослідження – комплекс розподілених систем, що знаходяться в загальній екосистемі, та проектування системи в рамках обраної екосистеми.

Предметною областю даної магістерської роботи є екосистема роздрібної торгівлі, в якій функціонує комплекс розподілених систем, що забезпечують успішність оплати товарів покупцями в мережі магазинів роздрібної торгівлі.

Новим результатом, отриманим у даній роботі, є розробка архітектури системи самообслуговування з самостійним скануванням, що значно покращує ефективність вже існуючих систем самообслуговування в області роздрібної торгівлі за рахунок розширення його можливостей та перенесення функціональності в мобільну площину. Розроблена архітектура дозволяє значно скоротити час на оплату та покращує рентабельність роботи мережі магазинів роздрібної торгівлі.

Результати роботи впроваджуються компанією ТОВ «ДІ ЕНД ДІ ГАРАНТ».

CRM, POS, POINT OF SALE, АРХІТЕКТУРА, ПРОЕКТУВАННЯ

## **ABSTRACT**

Graduation work of educational qualification level "?" on " Automated payment control system for retail Point of Sales (POS)". The work contains ? report pages, 11 sections, 29 illustrations, 18 tables and 8 appendices.

The purpose of the work – development of a self-service technology in the retail chain of stores, that would bring a huge time economy to the end user, increase revenue for the chain of stores itself and open a new way of targeted communication with the end user.

Subject of investigation – design of distributed software based on the existing architecture.

Research object – set of computing systems that are part of one ecosystem and design of a new system within the chosen ecosystem.

The subject area of this master's work is the ecosystem of retail trade, in which there is a complex of distributed systems that ensure the success of paying customers for goods in a network of retail stores.

The new result obtained in this work is the development of self-service system architecture with self-scanning, which significantly improves the efficiency of existing self-service point of sale systems in the retail field. Such improvement is possible due to extension of the possibilities and move part of functionality to mobile. Designed architecture allows to noticeably decrease time of sale and improve the profitability for chain of retail stores.

The results of the work are implemented by DI & D GARANT LLC.

CRM, POS, POINT OF SALE, ARCHITECTURE, DESIGN

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	9
ВСТУП .....	11
1.1 Загальна характеристика предметної області дослідження. ....	11
1.1.1. Головні тенденції розвитку систем роздрібної торгівлі .....	12
1.2.1. Каси самообслуговування.....	15
1.2.2. Відокремлене сканування .....	17
1.2.3. Самостійне сканування .....	18
1.2. Порівняльна оцінка існуючих систем самообслуговування.....	19
1.3. Мета роботи .....	21
2. ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ.....	22
2.1. Визначення зацікавлених сторін .....	22
2.2. Визначення вимог .....	23
2.3. Функціональні вимоги .....	24
2.4. Нефункціональні вимоги.....	25
3. СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ .....	27
3.1. Вхід в систему .....	27
3.2. Реєстрація в системі.....	28
3.3. Сканування товару .....	29
3.4. Визначення ціни товару.....	30
3.5. Створення нового кошика .....	31
3.6. Наповнення кошику .....	32
3.7. Редагування кошику .....	33
3.8. Керування лімітами.....	34

3.9.	Завершення покупки .....	35
3.10.	Сценарії взаємодії оператора .....	36
3.11.	Сценарії взаємодії адміністратора.....	37
4.	СТРУКТУРНА СХЕМА СИСТЕМИ.....	39
4.1.	Сфери роботи роздрібної торгівлі .....	39
4.2.	Схема системи управління роздрібною торгівлі .....	44
4.3.	Бізнес процес покупки .....	47
4.4.	Структурна схема системи самостійного сканування.....	49
5.	ВИБІР ТА ОБГРУНТУВАННЯ ЕЛЕМЕНТІВ ТА ТЕХНОЛОГІЙ .....	51
5.1.	Вибір архітектурного підходу.....	51
5.2.	Вибір технологій сервісно-орієнтованої архітектури .....	55
5.3.	Обґрунтування обраних технологій.....	57
5.4.	Вимоги до бази даних .....	58
5.5.	Обґрунтування вибору бази даних.....	59
6.	ER-ДІАГРАМА.....	61
6.1.	Модель бази даних .....	61
6.2.	Схема бази даних .....	62
7.	РЕАЛІЗАЦІЯ БІЗНЕС-ЛОГІКИ СИСТЕМИ .....	69
7.1.	Архітектура програмного продукту .....	69
7.1.1.	Рівень даних та доступу до них.....	71
7.1.2.	Рівень бізнес логіки .....	73
7.1.3.	Сервісний та презентаційний рівні .....	74
7.1.4.	Архітектура мобільного прикладення .....	75
7.2.	Діаграми послідовності .....	78

7.3.	Протоколи HTTP та HTTPS .....	81
7.4.	Інтеграція з Google ML (Machine Learning) Kit .....	84
8.	РОЗРОБЛЕННЯ ІНТЕРФЕЙСА КОРИСТУВАЧА.....	86
10	СТАРТАП-ПРОЕКТ.....	91
1.1	Загальний стан ринкового середовища для впровадження проекту.....	91
1.2	Опис ідеї проекту .....	93
1.3	Порівняння проекту з існуючими аналогами.....	95
1.4	Технологічний аудит проекту.....	97
1.5	Аналіз ринкових можливостей запуску стартап-проекту і стратегія виходу на ринок.....	99
1.6	Розробка маркетингової програми. ....	100
1.6.1	Формування маркетингової концепції товару .....	100
1.6.2	Конкурентна модель товару .....	101
1.7	Висновки .....	104
	ВИСНОВКИ.....	105
	Перелік використаних джерел .....	107



## ПЕРЕЛІК СКОРОЧЕНЬ

- ACID (*англ. Atomicity, Consistency, Isolation, Durability*) - набір властивостей, що гарантують надійну роботу транзакцій бази даних: атомарність, узгодженість, ізолюваність, довговічність
- API (*англ. Application Programming Interface*) - прикладний програмний інтерфейс
- CRM (*англ. Customer Relationship Management*) - система управління відносинами з клієнтами)
- CSS (*англ. Cascading Style Sheets*) - каскадна або блочна верстка веб-сторінок.
- DAO (*англ. Data Access Object*) - об'єкт доступу до даних
- ER-ДІАГРАМА (*англ. Entity-Relationship model*) - модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків
- ERP (*англ. Enterprise Resource Planning*) - система корпоративного планування ресурсів
- GWT (*англ. Google Web Toolkit*) - це Java фреймворк з відкритим кодом
- HRMS (*англ. Human Resource Management System*) - система управління людськими ресурсами
- HTML (*англ. Hypertext Markup Language*) - мова розмітки гіпертекстових документів
- HTTP (*англ. Hyper Text Transfer Protocol*) - протокол передачі даних
- ID (*англ. Identity Document*) - унікальна ознака об'єкта, ідентифікатор
- iOS - мобільна операційна система від Apple
- INT (*англ. Integer*) - цілочислений тип даних
- JavaFX - платформа та набір інструментів для створення інтернет додатків з можливістю підвантаження медіа та зміст
- JPA (*англ. Java Persistence API*) - стандартизований інтерфейс для Java ORM фреймворків
- JSF (*англ. Java Server Faces*) - це каркас веб-застосунків написаний на Java для розробки користувацьких інтерфейсів

JSON (*англ. JavaScript Object Notation*) - текстовий формат обміну даними між комп'ютерами.

JSP (*англ. Java Server Pages*) - технологія, що дозволяє веб-розробникам динамічно генерувати HTML, XML та інші веб-сторінки

MIS (*англ. Management Information System*) - інформаційна система управління

ORM (*англ. Object-Relational Mapping*) - об'єктно-реляційна проєкція

QR-код (*англ. Quick Response* - швидкий відгук) - матричний код (двовимірний штрих-код)

REST (*англ. Representational State Transfer*) - передача репрезентативного стану

RPC (*англ. Remote Procedure Call*) - виклик віддалених процедур

SCM (*англ. Supply Chain Management*) - управління ланцюгом поставок

SHA256 (*англ. Secure Hash Algorithm*) - безпечний алгоритм хешування

SOA (*англ. Service-Oriented Architecture*) - сервіс-орієнтована архітектура

SOAP (*англ. Simple Object Access Protocol*) - протокол обміну структурованими повідомленнями в розподілених обчислювальних системах

SQL (*англ. Structured Query Language*) - мова структурованих запитів для взаємодії користувача з базами даних

UI (*англ. User Interface*) - інтерфейс користувача

UML (*англ. Unified Modeling Language*) - уніфікована мова моделювання

UUID (*англ. Universally Unique Identifier*) - унікальний ідентифікатор користувача

Web (*англ. web* , «павутина» ) - інтернет-простір

## **ВСТУП**

### **1.1 Загальна характеристика предметної області дослідження.**

Ми живемо в епоху розвинутого товарного виробництва, історія якого сягає глибини тисячоліть. Пройшовши у своєму розвитку шлях від простого натурального до розширеного нематеріального товарне виробництво було нерозривно пов'язане з процесами обміну результатами праці - товарами. Після того, як з'явився особливий товар гроші, виникли товарно-грошові відносини і домінуючою формою товарного обміну стала торгівля - купівля-продаж, у ході якої товар обмінювався на його вартісний еквівалент, певну кількість грошей. Покупець передавав продавцеві гроші і продавець передавав покупцеві товар, за який отримав грошову оплату.

Розвиток торгівельних відносин нерозривно пов'язаний із суспільним прогресом. Цей зв'язок проявлявся як в зміні предметів торгівлі, так і в формах і технологіях здійснення торгових операцій. Особливо швидко ці зміни стали відбуватись в епоху охоплення інформаційними технологіями всіх сфер суспільного життя.

Інформаційні технології у сфері торгівлі не тільки змінили сам характер процесу купівлі-продажу, але через зворотній зв'язок створили небачені раніше можливості для товарного виробництва заповнювати ринок новими товарами і отримувати стимули для появи нових товарів із запрограмованими властивостями.

Для покупців інформатизація торгівлі відкрила можливість оперативно орієнтуватись серед мільйонів найменувань товарів, і з мінімальними витратами часу отримувати бажану річ без огляду на тисячі кілометрів які можуть фізично її віддаляти.

Кінцевою метою процесу торгівлі є продаж товару. Щоб знайти покупця для виставленого на продаж товару створюється ціла екосистема, яка працює над поширенням інформації про унікальні якості товару і пов'язані з ними нові можливості для його власника, яка створює максимально комфортні умови для безпосереднього ознайомлення покупця із зразками виставлених на продаж товарів і сприяє формуванню у нього бажання придбати товар та ще не раз повернутись саме в це місце за новими товарами після вдалої покупки.

Досягнути свою мету торгівля в сучасних умовах може лише опершись на можливості сучасних інформаційних технологій, особливо в самому масовому її секторі - роздрібному.

### **1.1. Головні тенденції розвитку систем роздрібної торгівлі**

Роздрібна торгівля (рітейл) є завершальною ланкою в ланцюжку руху товарів від виробника до кінцевого споживача. В розрізі нашої роботи, важливо відзначити такі її особливості:

- В торгівельних відносинах задіяна велика кількість та різноманітність товарів, які є предметами купівлі-продажу.
- Наявність величезної кількості видів та номенклатури товарів вимагає розгортати процес продажу у спеціально облаштованих торгових приміщеннях, в яких покупцям створюється можливість придбати тисячі товарів.
- Кожна подія купівлі-продажу, не зважаючи на різноманітність предметів продажу, має типовий характер і відвідування покупцем роздрібного магазину можна описати типовим алгоритмом з кількома вкладеними циклами, який починається тим, що покупець входить у торговий зал, а закінчується його розрахунком на касі за відібраний у залі товар.

Традиційним методом організації роздрібного продажу є «продаж через прилавок». Алгоритм продажу товару таким методом наведений на рис. 1.1 [ ].

Описаний метод продажу товару не зважаючи на тисячолітню історію зберігся до наших днів і з ним можна зустрітись, зокрема, не тільки у випадках, коли мова йде про продаж невеликої номенклатури штучних (ексклюзивних) товарів, а також тоді, коли інші способи організації торгівлі є економічно недоцільними, наприклад, на роздрібному ринку, коли продавців товару більше ніж його покупців..

Для «продажу через прилавок» характерне максимальне питоме залучення торгового персоналу у процес купівлі-продажу: покупці обслуговуються продавцем по

черзі і збільшення пропускної здатності торгового залу можливе лише за рахунок організації ще однієї паралельної черги із своїм продавцем. Також для цього типу продажу характерні фізичні обмеження у просторі для розташування товарних пропозицій і, як результат, у прийнятному часі на обслуговування покупця.

Прогрес у сфері організації роздрібної торгівлі йшов, з одного боку, по шляху створення для покупця більш комфортних умови для вибору товару і прийняття рішення про його придбання та скорочення витрати часу на здійснення покупки, а з іншого боку, для торгового закладу, був спрямований на збільшення пропускної спроможності торгового залу та зменшення питомих видатків на продаж одиниці товару.

Типовим методом організації роздрібного продажу товарів у наш час є «самообслуговування», яке реалізоване у переважній більшості, так званих, супер(гіпер)-маркетів.

Алгоритм продажу товару таким методом наведений на рис. 1-2 [ 1 ].

Відмінність алгоритму «самообслуговування» від алгоритму «продаж через прилавки» підвищує ефективність продаж за рахунок того, що всі дії по підборі товару, їх доставку до місця розрахунку та їх упаковку покупець виконує самотужки, завдяки чому алгоритм продажу стає паралельним у всіх частинах, окрім розрахункових операцій, які виконуються касиром. Саме касові вузли стають вузьким місцем для магазину, особливо в часи пікового потоку покупців - довге очікування на касі викликає у покупців невдоволення та роздратування, а регулярні втрати часу на касах сприяють зниженню лояльності покупців до конкретної торгової мережі. Кращий фінансовий результат отримують ті мережі, які забезпечують високий рівень обслуговування у торговому залі (зручне розташування товару, продумана інформаційна підтримка процесу пошуку товару, яка покращує орієнтацію покупців у торговому залі, логічне групування товарів, яке робить осмисленим маршрут пересування покупця по торговому залу для наповнення кошика покупок,.. тощо) у поєднанні з можливістю швидко розрахуватись на касі, що прямо залежить від кількості кас. Сьогодні

рітейлу для підтримки конкурентоздатності уже недостатньо просто мати вдалий асортимент товарів, зробити їх доступними в сучасному торговельному залі самообслуговування і запропонувати привабливі ціни на товари. Конкуренція на ринку вимагає від рітейлера вибору найбільш економічно ефективної організації процесу продаж. В свою чергу клієнти стають більш вимогливими до рівня обслуговування.

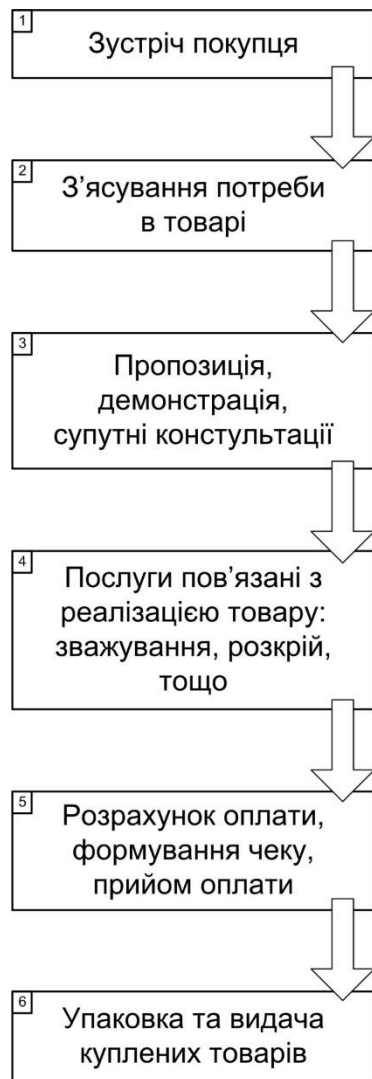


Рис. 1.1

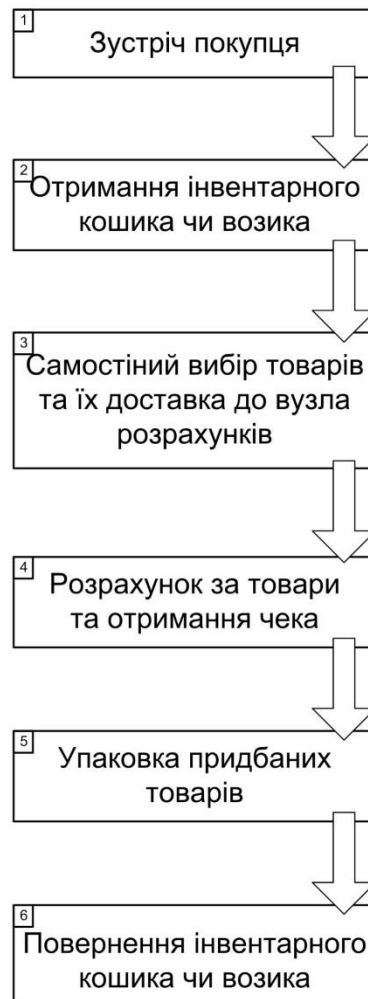


Рис. 1-2

Подальший розвиток систем роздрібного продажу був спрямований на оптимізацію блоку розрахунку за обраний покупцем товар шляхом впровадження елементів самообслуговування при формуванні чеку та виконанні оплати.

Станом на сьогодні сформувались основні напрямки впровадження самообслуговування покупців при оплаті за товари [ ]:

- Каси самообслуговування (Self-Checkout)
- Відокремлене сканування (Scan & Pay)
- Самостійне сканування (Self Scanning)

Розглянемо їхні характерні риси.

#### **1.2.1. Каси самообслуговування**

У 2013 році світовий досвід подальшого підвищення ефективності організації процесу торгівлі, а саме, впровадження кас самообслуговування перетнув кордони України і досягнув Києва - у одному з супермаркетів «Велика Кишеня» була встановлена така каса. Каси самообслуговування дозволяють «розпаралелити» процеси в блоці розрахунку за товар алгоритму продажу.

Каси самообслуговування - це комплекси технічних засобів, що дозволяють покупцеві самостійно сканувати промарковані штрих-кодами товари, при потребі, зважувати товар та виконувати оплату. До складу типового апаратно-програмного комплексу засобів, яким є каса самообслуговування, входять [ ]:

- системний блок каси для виконання на ньому стандартної фронт-офісної програми рітейлера;
- монітора з сенсорним екраном для інтерактивної взаємодії з покупцем;
- сканера чи сканер-ваг для сканування штрих-коду товару;
- контрольних ваг у платформі для пакетів;
- акцептори (для прийому) і диспенсери (для видачі решти) купюр та монет;
- касетних ящиків для зберігання купюр та монет;

- приладу для дезактивації протикрадіжних маркерів;
- системи звукового і світлового оповіщення;
- авторизаційного терміналу для обслуговування платіжних карт
- фіскального реєстратора–принтера для друку чеків;

Робота з касою самообслуговування містить всі операції, які виконує касир у звичайному магазині самообслуговування, лише з тією відмінністю, що їх виконує не працівник магазину, а сам покупець. Ефективність кас самообслуговування визначають такі головні чинники:

- Не потрібна присутність на кожній касі працівника магазину, касир-консультант може надавати підказки покупцям, які самостійно сканують відібрані товари, на 5 - 10 касах самообслуговування.
- Використання кас самообслуговування дозволяє економити торгову площу - каса самообслуговування займає в 2 - 3 рази меншу площу торгового залу у порівнянні з класичною касою, що дозволяє звільнити додаткову площу під розташування товару без зниження пропускної спроможності торгового залу.

При використанні кас самообслуговування персонал магазину не залучений до роботи з готівкою і сканером, всі операції, що здійснюються покупцем, контролюються неупередженою машиною. Також, у порівнянні із звичайною касою, готівка обробляється у повністю автоматичному режимі, який усуває людський фактор та виключає прийом в оплату фальшивих банкнот. Окрім цього, використання кас самообслуговування сприяє зменшенню кількості крадіжок, які часто здійснюються за участю персоналу саме на касах в результаті змови недобросовісних покупців з касиром.

Технологія кас самообслуговування має безсумнівні переваги перед класичними касами, але практика використання таких кас підказала шляхи подальшого розвитку самообслуговування. Можливість її подальшої вдосконалення визначається самим характером процедури розрахунку, яка складається з двох фаз: формування



чеку (послідовного сканування одиниць товару) і розрахунків за товар. Аналіз внеску кожної з цих фаз у результуючі витрати часу на вихід з торгового залу [ ] показує, що більше половини часу при невеликій кількості одиниць товару покупець витрачає на оплату покупки. У випадку перевищення числом одиниць товару певної кількості, яка ще вільно розташовується на робочих поверхнях касового вузла, різко сповільнюється процес сканування покупцем одиниць товару через фізичну незручність оперування з товарними одиницями. яка нашаровується на непрофесіоналізм покупця при виконанні роботи касира. Логічним кроком до підвищення пропускної здатності вузла розрахунку стало відокремлення операцій сканування товарів від операції розрахунку за товар. При цьому сканування доручається виконувати професіоналу, а розраховується за товар покупець самостійно. При цьому для розширення вузького місця облаштовують більшу кількість розрахункових терміналів ніж кількість ліній сканування. Нова технологія самообслуговування при розрахунку за покупки отримала назву «відокремлене сканування».

### **1.2.2. Відокремлене сканування**

Особливістю технології відокремленого сканування є виконання, на перший погляд, ніби кроку назад - повернення до сканування товарних одиниць персоналом магазину. Співробітник магазину не приймає оплату від покупця, а лише сканує товарні одиниці і формує попередній чек, за яким покупець самостійно розраховується на терміналі оплати.

Зона сканування складається зі станцій, схожих на звичайні касові вузли, обладнаних продуктивними сканерами штрих-кодів, що дозволяють співробітникам магазину ефективно сканувати покупки клієнта. Після сканування покупцеві видається попередній чек для подальшої самостійної оплати.

Зона оплати складається з автоматичних станцій оплати, і проектується таким чином, щоб кількість терміналів оплати було значно більше числа ліній сканування і дозволяло уникати черг на цьому етапі. Покупець оплачує товари тим способом, який йому зручний - з допомогою готівки, банківської карти або використовуючи змішану

оплату. Після оплати покупки він разом з касовим чеком отримує купон для виходу через автоматичні ворота, які відкриваються після піднесення до зчитувача отриманого при оплаті купона з QR-кодом. Такий підхід дозволяє забезпечити контроль оплати відсканованих покупок і влаштувати організований вихід з торгового залу.

Використання підходу відокремленого сканування показує вищу ефективність у порівнянні з касами самообслуговування при великому потоці покупців, але повернення до залучення співробітників магазину до сканування товарних одиниць знову встановлює межу для росту пропускної спроможності торгового залу. Спробою подолати це обмеження є розробка технології самотійного сканування при здійсненні покупок.

### **1.2.3. Самостійне сканування**

Технологія самотійного сканування [ ] організована таким чином: учасники програми лояльності рітейлера, заходячи в торговий зал, отримують персональний сканер. З його допомогою вони сканують обрані товари перед тим як покласти їх у кошик. При цьому сканер послідовно формує список покупок - основу розрахункового чека у майбутньому. Коли покупець завершує наповнювати кошик покупками, він підходить до вузла розрахунку, підносить персональний сканер до зчитувача касового апарату і натискає кнопку «Оплатити». Касовий апарат отримує із сканера список покупок, приймає від покупця оплату і друкує чек.

Для того, щоб рітейлер був упевнений в тому, що всі товари в кошику відскановані покупцем, в системі передбачені різні алгоритми, які аналізують дії людини в торговому залі. Таких контрольованих параметрів можуть бути десятки - налаштовує їх сам рітейлер. У разі, якщо система виявляє підозрілі дії, покупець потрапляє під перевірку: часткове або повне сканування товарів з кошика. Також можливо ручне призначення перевірки співробітниками служби безпеки. Вихід із зони оплати додатково контролюється аналогічно тому, як це відбувається в системі відокремленого сканування.

Продаж товарів з використанням самостійного сканування у порівнянні з розглянутими раніше методами самообслуговування при оплаті за товар дозволяє мінімізувати черги на виході з торгового залу, але його впровадження супроводжується значними витратами на придбання портативних сканерів, на їхнє обслуговування, ремонт, та втрати, що неминучі при непрофесійній експлуатації прецизійних електронних приладів. Саме ці супутні витрати знижують привабливість такої технології самообслуговування.

## 1.2. Порівняльна оцінка існуючих систем самообслуговування

Порівняльні характеристики існуючих систем самообслуговування наведені нижче, в таблиці 1-1.

Таблиця 1-1

п/п	Назва	Очікування в черзі	Тривалість сканування	Зручність	Швидкість розрахунку за покупки	Загальна сума балів
	Каса самообслуговування	1	1	1	1	3
	Відокремлене сканування	1	2	2	1	6
	Самостійне сканування	2	3	3	1	9

У таблиці 1-1 показано умовне нарахування балів, які визначаються якісними властивостями кожної системи, щодо зручності користування та швидкості розрахунку і виходу за межі торгівельного закладу, зокрема:

- Очікування в черзі на обслуговування практично однакове при використанні кас самообслуговування та при відокремленому сканування. При самотійному скануванні покупець підходить до вузла розрахунку із сформованим списком покупок, що знаходяться у нього в кошику, і дії, які він має виконати для оплати за товари в кошику - це передача списку на касу та внесення грошових коштів. Технологія самотійного сканування не потребує наявності на касі засобів сканування товарів, бо всі товари вже відскановані покупцем при їх переміщенні з магазинної полиці у кошик.
- Тривалість сканування товарів при використанні кас самообслуговування більша ніж при використанні відокремленого сканування через незручності, пов'язані з необхідністю поводження із значною кількістю товарів на обмеженому просторі касового вузла, та через непрофесійність виконання цих операцій покупцем у порівнянні з персоналом магазину, який вправніше виконує ці типові операції.
- Зручність використання кас самообслуговування нижча ніж у випадку виконання операцій сканування персоналом магазину. Ще більш зручним є використання самотійного сканування, при якому немає потреби діставати товари із кошика і виконувати сканування - при підході до касового вузла весь товар уже відсканований.
- Швидкість розрахунку за покупки для всіх способів самообслуговування практично однакова, бо зводиться до типових дій: піднесення банківської картки до POS-терміналу і введення PIN-коду, чи до внесення коштів у термінал для банкнот і отримання решти.

За загальною сумою балів найбільш ефективним способом самообслуговування є самостійне сканування і єдиним недоліком, який знижує привабливість цієї технології є значні витрати на придбання портативних сканерів, на їхнє обслуговування, ремонт, та втрати, що неминучі при непрофесійній експлуатації прецизійних електронних приладів.

### **1.3. Мета роботи**

Метою цієї роботи є розробка технології самообслуговування покупців, яка вільна від недоліків способу самообслуговування з самостійним скануванням.

Сформульовану мету передбачається досягти шляхом розробки програмного додатку “ПОКУПЕЦЬ” для смартфона, який дасть можливість покупцю використувати смартфон у якості персонального сканера аналогічно до способу самообслуговування «самостійне сканування». За допомогою смартфона покупець зможе сканувати кодові мітки на етикетках обраних товарів перед тим як покласти їх у кошик. Після завершення наповнення кошика з покупками покупець на касовому вузлі зможе передати сформований список в касовий апарат для розрахунку загальної вартості всіх товарів у кошику і формування чека на оплату та розраховується за покупки.

## **2. ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ**

Формування вимог до системи це багатоетапний та складний процес, що є чи не найважливішим у побудові будь-якої системи. Від успіху цього етапу залежить наскільки точно створена та впроваджена система відповідатиме очікуванням зацікавлених сторін.

Першою стадією формування вимог має бути визначення зацікавлених сторін та виявлення вигоди. Потрібно зауважити, що не всі зацікавлені сторони можуть отримувати вигоду у економічному, тобто фінансовому, вигляді. Вигода також може бути політичною, функціональною та соціальною. Наприклад, спрощення бізнес процесу, зменшення кількості складних інтеракції із системою, зменшення витраченого на навчання та на взаємодію часу також є вагомими факторами, які можуть і не приносити прямої фінансової вигоди, проте можуть позитивно впливати на лояльність працівників, клієнтів, репутацію, а значить, і на кінцеве рішення щодо початку проектування, а надалі і впровадження, подібної системи.

### **2.1. Визначення зацікавлених сторін**

Після проведення огляду існуючих рішень було однозначно визначено, що перша зацікавлена сторона – мережі роздрібної торгівлі.

Сформулюємо фактори, що впливають на економічну вигоду:

- 1) Зниження витрат на придбання портативних сканерів
- 2) Зниження витрат на обслуговування портативних сканерів
- 3) Зниження витрат на ремонт портативних сканерів
- 4) Можливість більш гнучкої та персоналізованої побудови рекламних кампанії

Фактори, що впливають на соціальну вигоду:

- 1) Покращення репутації магазину
- 2) Збільшення кількості лояльних клієнтів

Другою зацікавленою стороною є покупець (клієнт мережі роздрібної торгівлі). Покупець в даному випадку не є отримувачем прямої економічної вигоди, проте він є отримувачем соціальної:

- 1) Зменшення часу, проведеного в магазині

По суті, весь час перебування в магазині зводиться до самого процесу вибору товарів, так як відсутній повторний перебір всіх товарів в кошику та їх сканування.

- 2) Забезпечення приватності покупки

Після визначення зацікавлених сторін, необхідно визначити вимоги кожної з цих сторін до проектуємої системи.

## **2.2.Визначення вимог**

Весь комплекс вимог до програмного додатку можна поділити на функціональні та не функціональні:

- 1) Функціональні вимоги

Описують поведінку системи у відповідності до її функціонального призначення і визначають весь обсяг операцій, обробку та маніпуляції даних, а також специфічні функції, які виконує система – як внутрішні, так і зовнішні. Джерелом функціональних вимог мають бути зацікавлені сторони, що взаємодіють із системою. Зазвичай, функціональні вимоги продиктовані в більшій мірі операціями, що має мати можливість здійснювати користувачі системи. Користувачами системи можуть бути представники різних зацікавлених сторін.

Функціональні вимоги відповідають на питання, як система повинна працювати?

- 2) Нефункціональні вимоги – описують критерії якості розробленої системи а також задають обмеження на допустимі дії. Основним джерелом нефункціональних вимог є бізнес.

Нефункціональні вимоги відповідають на питання, якою система повинна бути?

В даній роботі, джерелами вимог є три типи користувачів, що є представниками різних зацікавлених сторін:

- 1) Покупець – клієнт мережі роздрібної торгівлі
- 2) Оператор – працівник мережі роздрібної торгівлі
- 3) Адміністратор – працівник мережі роздрібної торгівлі

### **2.3. Функціональні вимоги**

Функціональні вимоги, джерелом яких є покупець:

- 1) Користувач повинен мати змогу увійти до системи за допомогою свого логіну та пароля
- 2) Користувач повинен мати змогу зареєструватись у додатку
- 3) Користувач повинен мати змогу за допомогою додатку просканувати одиницю товару
- 4) Користувач повинен мати змогу за допомогою додатку визначати актуальну ціну одиниці товару
- 5) Користувач повинен мати змогу створити новий кошик в системі
- 6) Користувач повинен мати змогу за допомогою додатку формувати список одиниць товару, які він має намір купити (наповнювати кошик товарам)
- 7) Користувач повинен мати змогу за допомогою додатку редагувати список покупок
- 8) Користувач повинен мати змогу за допомогою додатку контролювати суму вартості зібраних у кошику товарів
- 9) Користувач повинен мати змогу за допомогою додатку передати список покупок на касовий вузол для розрахунку
- 10) Користувач повинен мати змогу завершити покупку переведенням статусу кошика в «Готовий до оплати»



- 11) Система має автоматично анулювати кошик користувача і відключитись від інформаційної системи магазину у разі, якщо користувач неактивний більше 30хв
- 12) Користувач повинен мати змогу через додаток поспілкуватися зі службою підтримки, в разі виникнення будь-яких питань
- 13) Користувач повинен мати змогу запросити допомогу оператора із ручним введенням одиниці товару до коширу в разі, якщо додаток не в стані розпізнати штрих- чи QR-код товару.

Функціональні вимоги, джерелом яких є адміністратор:

- 1) Користувач має мати змогу додати будь-який товар в систему
- 2) Користувач має мати змогу змінити найменування, опис, вагу чи інші характеристики будь-якого товару в системі
- 3) Користувач має мати змогу видалити будь-який товар в системі
- 4) Користувач має мати змогу внести в систему похибку, що використовується при підтвердженні кошика через його зважування

Функціональні вимоги, джерелом яких є оператор:

- 1) Користувач має мати змогу відповісти на запит покупця та підтвердити ручне введення одиниці товару в разі неможливості його сканування
- 2) Користувач має мати змогу відповісти на будь-яке питання покупця в разі його виникнення через додаток
- 3) Користувач має мати змогу отримати запит на допомогу від покупця біля каси самообслуговування

## **2.4. Нефункціональні вимоги**

- 1) Система повинна забезпечувати приватність даних при операціях реєстрації та обміну даними з інформаційними ресурсами магазину
- 2) Система повинна забезпечувати можливість внесення у список покупок до 1000 одиниць товару

- 3) Система повинна підтримувати одночасно до 10000 активних кошиків
- 4) Система повинна забезпечувати збереження списку при втраті з'єднання з інформаційними ресурсами магазину
- 5) Система повинна забезпечувати автоматичне відновлення з'єднання з інформаційними ресурсами магазину
- 6) Програмна реалізація системи не повинна бути прив'язана до конкретної операційної системи.

### 3. СЦЕНАРІЙ ВИКОРИСТАННЯ СИСТЕМИ

Після визначення функціональних та нефункціональних вимог, необхідно побудувати сценарії використання системи, які будуть відображати визначені вимоги та детально описувати їх реалізацію.

Сценарієм використання системи є фактично прописана послідовність зовнішніх дій та запитів, які можна проводити в системі. Ця стадія аналізу допомагає визначати властивості нової системи та наглядно їх демонструє.

#### 3.1. Вхід в систему

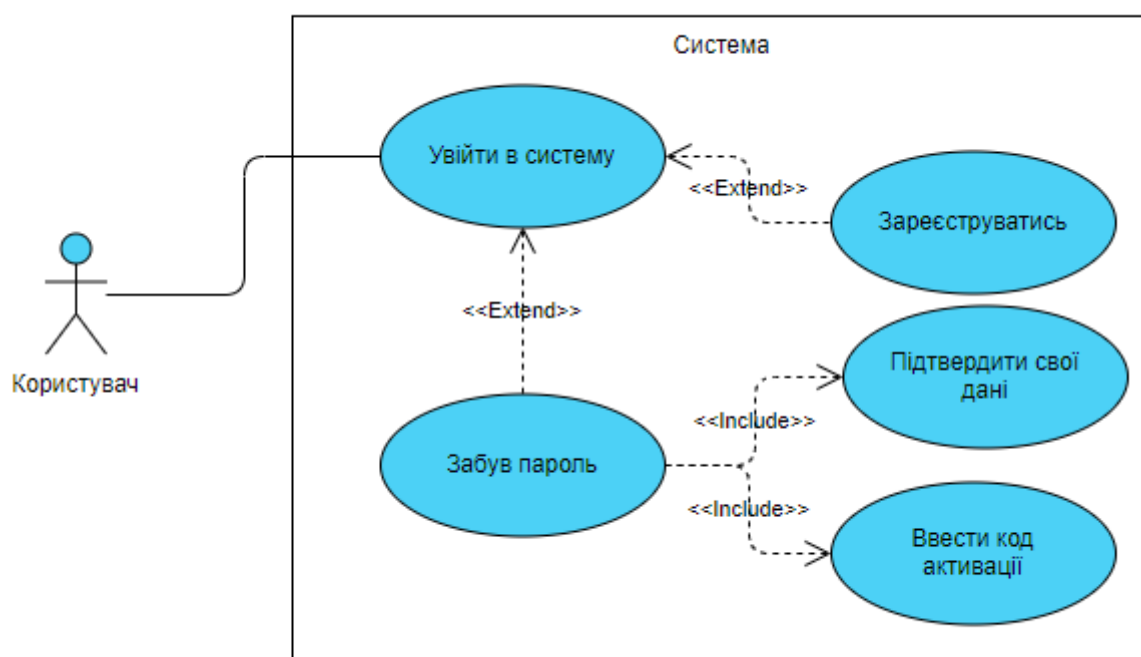


Рисунок 2 – UML діаграма сценарію входу в систему

Для використання смартфона у якості інструменту для здійснення покупок користувачеві необхідно запустити програму додатку на своєму смартфоні.

Після запуску, користувачеві буде запропоновано увійти в системи за допомогою введення логіну та паролю. Якщо користувач не був попередньо зареєстрований у системі, йому буде запропоновано крок реєстрації, де користувач зможе зареєструватись. Реєстрація обов'язково включає в себе введення коду активації для підтвердження свого аккаунту.

У разі, якщо користувач забув свій пароль, система запропонує йому обрати спосіб відновлення паролю – за допомогою мобільного номеру, чи електронної пошти, та надішле на вказану користувачем адресу код активації у випадку, якщо введена інформація співпала з наявною у базі даних. Після введення правильного коду активації, користувач матиме можливість обрати новий пароль для свого аккаунту.

Після успішного входу в системи, користувач отримує доступ до повного функціоналу здійснення покупки.

### 3.2. Реєстрація в системі

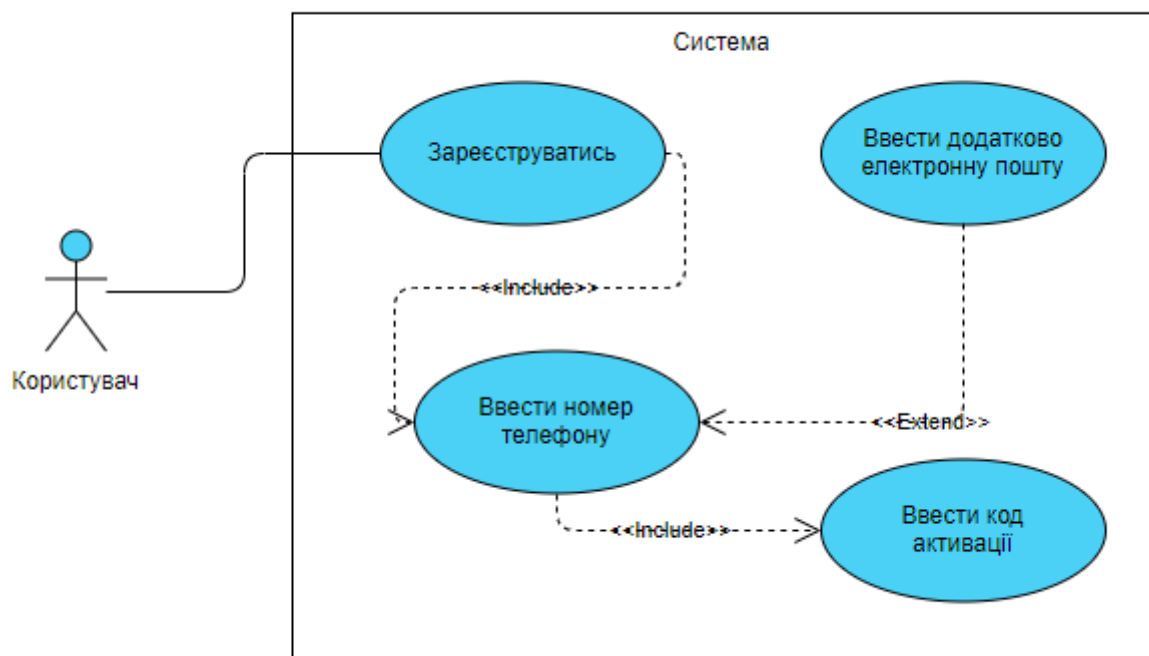


Рисунок 3 – UML діаграма сценарію реєстрації в системі

Для використання смартфона у якості інструменту для здійснення покупок користувачеві необхідно виконати підготовчі дії:

- запустити програму додатку на своєму смартфоні
- виконати початкову реєстрацію

При запуску додатку на смартфоні користувач може обрати опцію реєстрації та створити свій персональний аккаунт шляхом введення свого мобільного телефону та

за бажанням електронної адреси, а також задати пароль, який буде використовуватись при перебуванні у магазині для підключення до системи самообслуговування.

Після підтвердження користувачем внесення електронної адреси та пароля йому на вказаний номер телефону, або на поштову адресу, надійде код активації, який необхідно буде ввести з клавіатури для завершення процедури реєстрації.

Після успішної реєстрації, користувач отримує доступ до повного функціоналу здійснення покупки.

### 3.3. Сканування товару

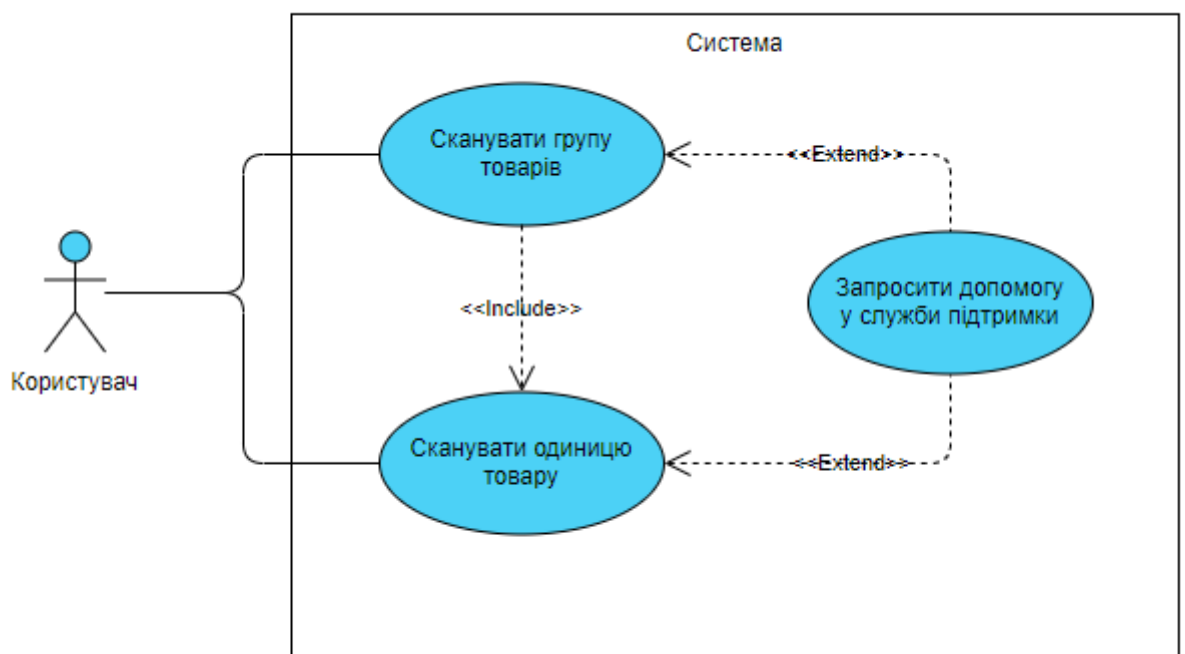


Рисунок 4 – UML діаграма сценарію сканування товару

Через інтерфейс камери, система надає користувачу можливість сканувати товари. Така можливість стає доступною лише після входження до системи, або ж після реєстрації в ній.

Увійшовши до системи, користувач може просканувати цілу групу товарів, відкривши сканер в додатку. Системою підтримується як сканування штрих-коду, так і сканування QR-коду.

Сканування групи товарів включає в себе сканування одиниці товару.

У разі виникнення проблем зі скануванням, користувач може запросити допомогу у служби підтримки через чат у прикладенні. В разі необхідності, він може вручну ввести код товару в системі. Ручне введення має бути підтверджене оператором системи, після чого у ручний спосіб введений товар з'явиться у користувача в кошику.

### 3.4. Визначення ціни товару

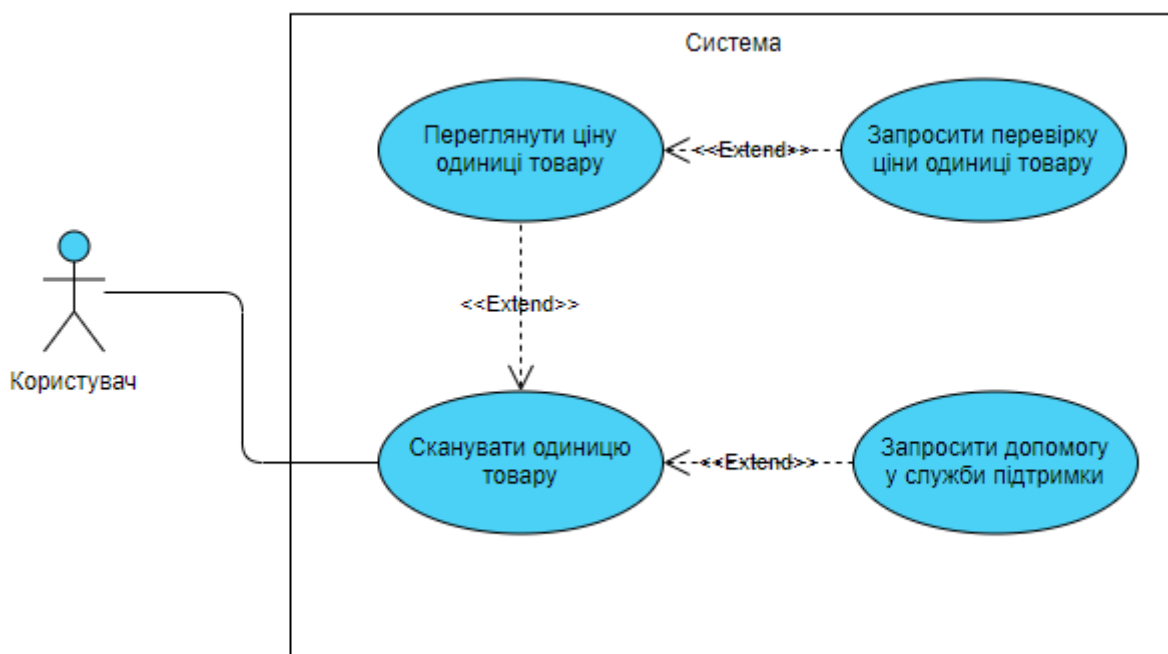


Рисунок 5 – UML діаграма сценарію визначення ціни товару

Після успішного входу в системи, або реєстрації в системі, користувач отримує доступ до функціоналу сканування товарів.

Якщо сканування одиниці товару було успішним, в спливаючому вікні користувач має опцію перегляду ціни на проскановану одиницю товару.

Ціна одиниці товару в системі може не співпадати з ціною цього ж товару на полиці магазину роздрібних торгівлі. В такому випадку, користувачу надається можливість запросити перевірку ціни на проскановану одиницю товару в служби

підтримки. Оператор має надати оперативну допомогу, та в разі існування подібної розбіжності, усунути її через модифікацію ціни в системі.

В разі, якщо сканування було неуспішним, користувач має можливість запросити допомоги у служби підтримки через внутрішній чат в додатку.

### 3.5. Створення нового кошика

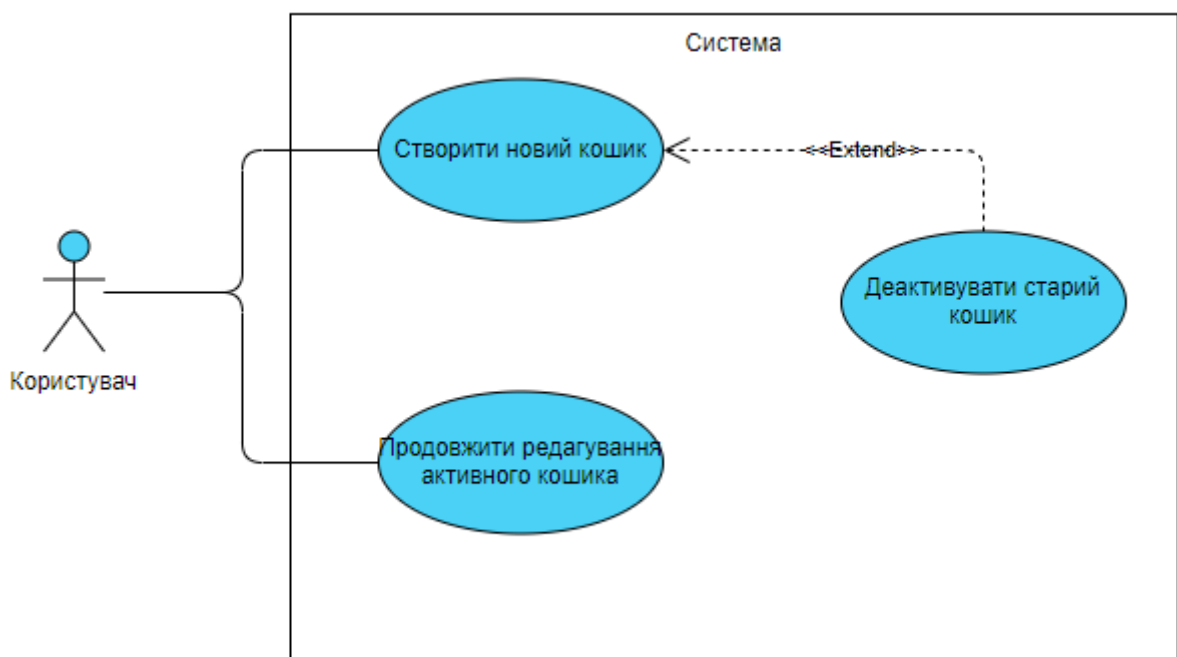


Рисунок 6 – UML діаграма сценарію створення кошика

Користувач, після успішного входження до системи або реєстрації, отримує можливість створення нового кошика.

Створення нового кошика можливе лише за умови, якщо у користувача немає існуючого активного кошика.

В разі, якщо наявний активний кошик, користувач має або деактивувати його та відмінити, або ж продовжити його редагування.

### 3.6. Наповнення кошику

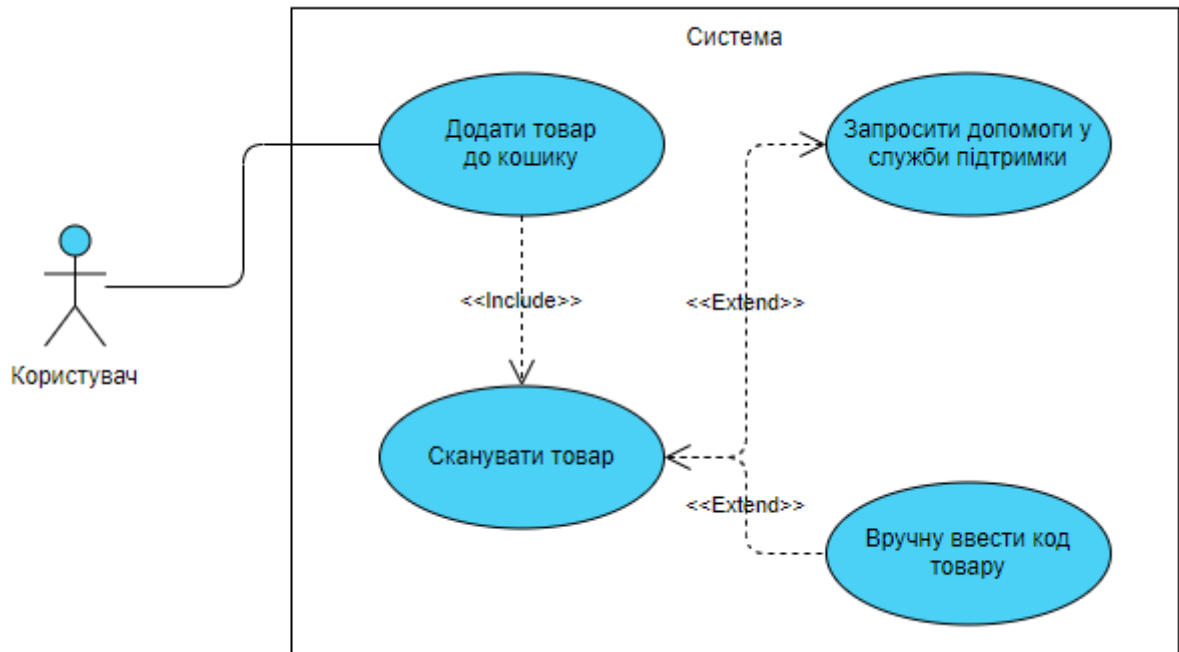


Рисунок 7 – UML діаграма сценарію наповнення кошику

Користувач, після успішного входу в системи, або ж після реєстрації, може створити свій персональний кошик та почати його наповнювати.

Наповнення кошику може відбуватися двома шляхами:

- сканування товару за допомогою вбудованого в додаток сканера;
- ручне введення коду товару.

У випадку успішного сканування одиниці товару, в спливаючому вікні разом з опцією перегляду ціни товару, стане доступною опція додавання товару до кошику.

Якщо сканування товару призводить до системної помилки, або ж неможливо за певних умов, користувачу надається можливість введення коду товару в ручному режимі.

Ручне введення товару має бути підтверджене оператором системи.

Також, при будь-якому виникненні помилки в додаванні товару до кошику, його скануванні, чи ручному введенні його коду, користувач має можливість звернутися в службу підтримки системи за допомогою.



### 3.7. Редагування кошику

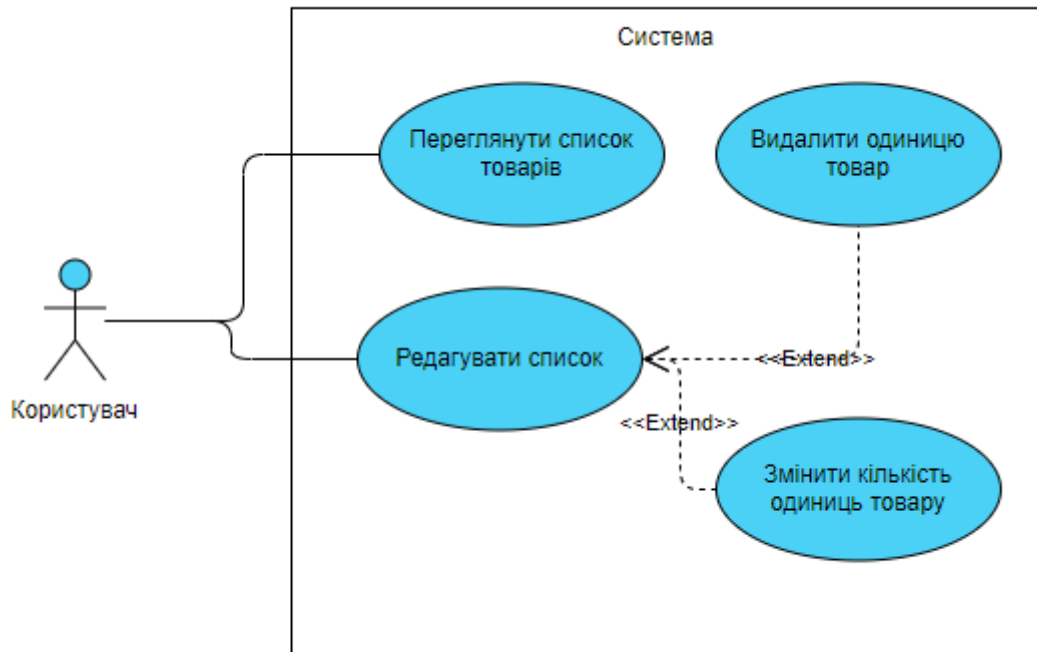


Рисунок 8 – UML діаграма сценарію редагування кошику

Користувач повинен мати змогу за допомогою додатку редагувати список покупок.

Редагування списку товарів передбачає такі дії:

Перегляд списку - для цього користувач повинен натиснути на екрані смартфона на кнопку «СПИСОК», після чого у інформаційному вікні на екрані смартфона з'являється список покладених у кошик товарів, в кожному рядку якого вказуються: назва товару, ціна за одиницю товару, кількість одиниць товару, покладених у кошик. Також в окремому вікні на екрані показується загальна вартість зібраних у кошику товарів.

Редагування списку - виконується окремо по кожній товарній позиції. В цьому режимі користувачеві доступні дві дії зі списком:

Змінити кількість одиниць товару - для переходу в режим зміни кількості користувач у режимі перегляду списку повинен двічі натиснути на рядок з товарною позицією, яку необхідно відредагувати.

Видалити товарну позицію - для цього користувач повинен посунути рядок з товарною позицією вліво чи вправо, після чого натиснути на кнопку «видалити» у вікні підтвердження операції.

### 3.8. Керування лімітами

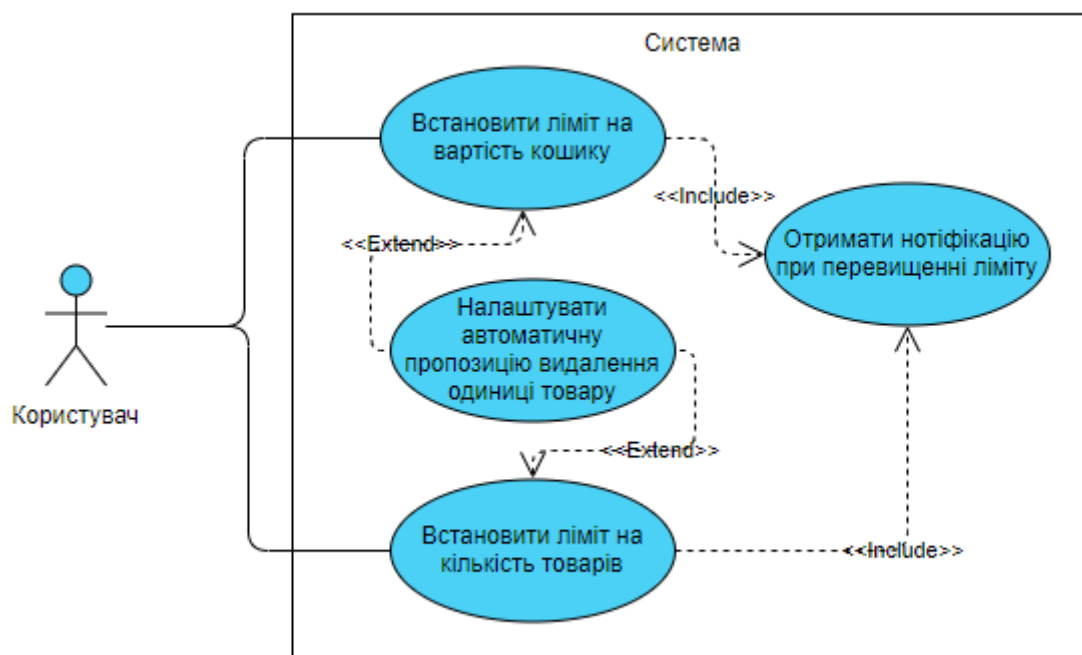


Рисунок 9 – UML діаграма сценарію керування лімітами  
Користувач має можливість встановлювати ліміти на свій кошик.  
Ліміти можуть бути двох видів:

- Ліміт на загальну вартість кошику
- Ліміт на кількість одиниць товарів в кошику

Ціль лімітів – допомогти користувачеві формувати бюджет на активний кошик. В разі перевищення ліміту, користувач отримає повідомлення у вигляді спливаючої нотифікації. Також, в разі перевищення межі, система запропонує користувачеві видалити самий (самі) дорогі, або ж самий (самі) дешеві товари з кошику, що дозво-

лить залишитись в межах встановленої загальної суми кошика, або ж кількості товарів. Подібна поведінка системи є опціональною та може задаватись користувачем в налаштуваннях.

### 3.9. Завершення покупки

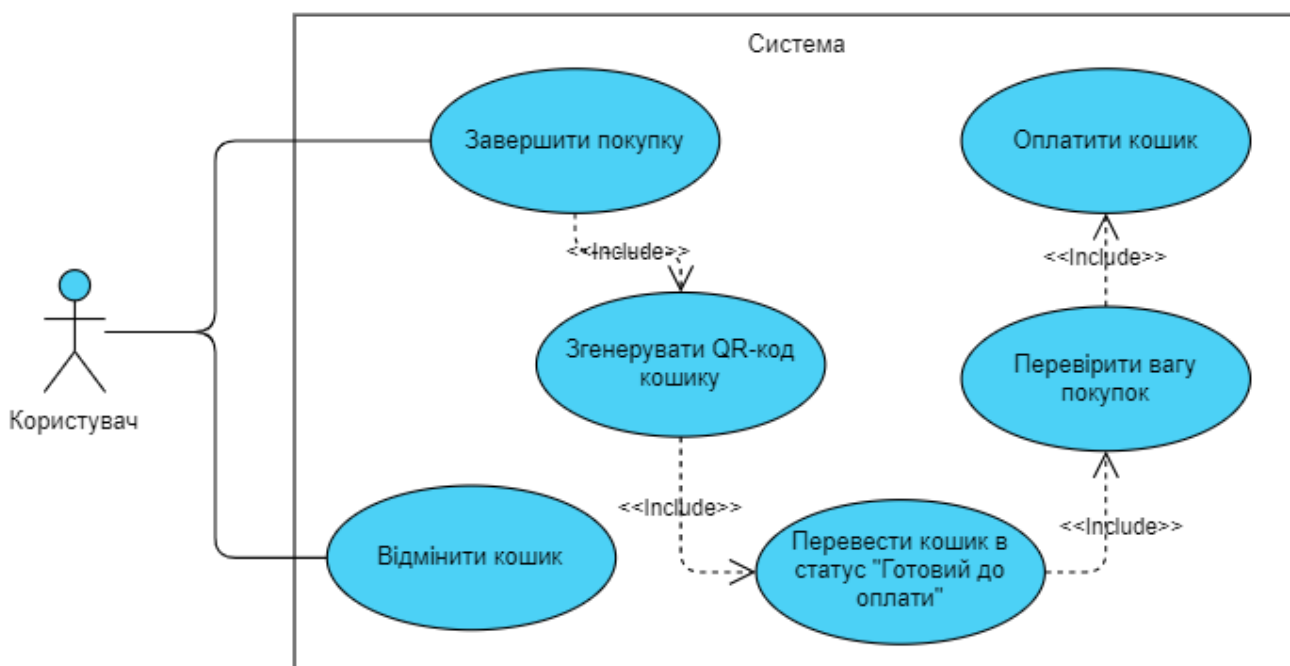


Рисунок 10 – UML діаграма завершення покупки

Користувач, після наповнення кошику товарами, має змогу здійснити наступні маніпуляції щодо кошика:

- завершити покупку;
- відмінити кошик.

Відміна кошика фактично деактивує наразі активний кошик у користувача та анулює всі додані до нього товари. Після деактивації, користувач має залишити всі товари, які він обрав в процесі покупки в пункті сортування.

Завершення кошика проходить в кілька етапів:

- Система генерує QR код для кошика після того, як користувач натискає на клавішу «Завершити покупку»

- Система запитує користувача, чи хоче він перевести кошик в статус «Готовий до оплати». Якщо користувач підтверджує готовність, система пропонує просканувати QR-код кошика на касі самообслуговування. Якщо користувач не підтверджує таку готовність, кошик продовжує перебувати в активному стані, а попередньо згенерований QR-код анулюється.
- Система пропонує користувачу зважити його товари задля перевірки загальної ваги покупки
- В разі, якщо вага підтверджується, система пропонує можливість до оплати кошика. Якщо вага не підтверджується, автоматично викликається оператор, що допомагає усунути розбіжність та підтверджує правильність кошику.

### 3.10. Сценарії взаємодії оператора

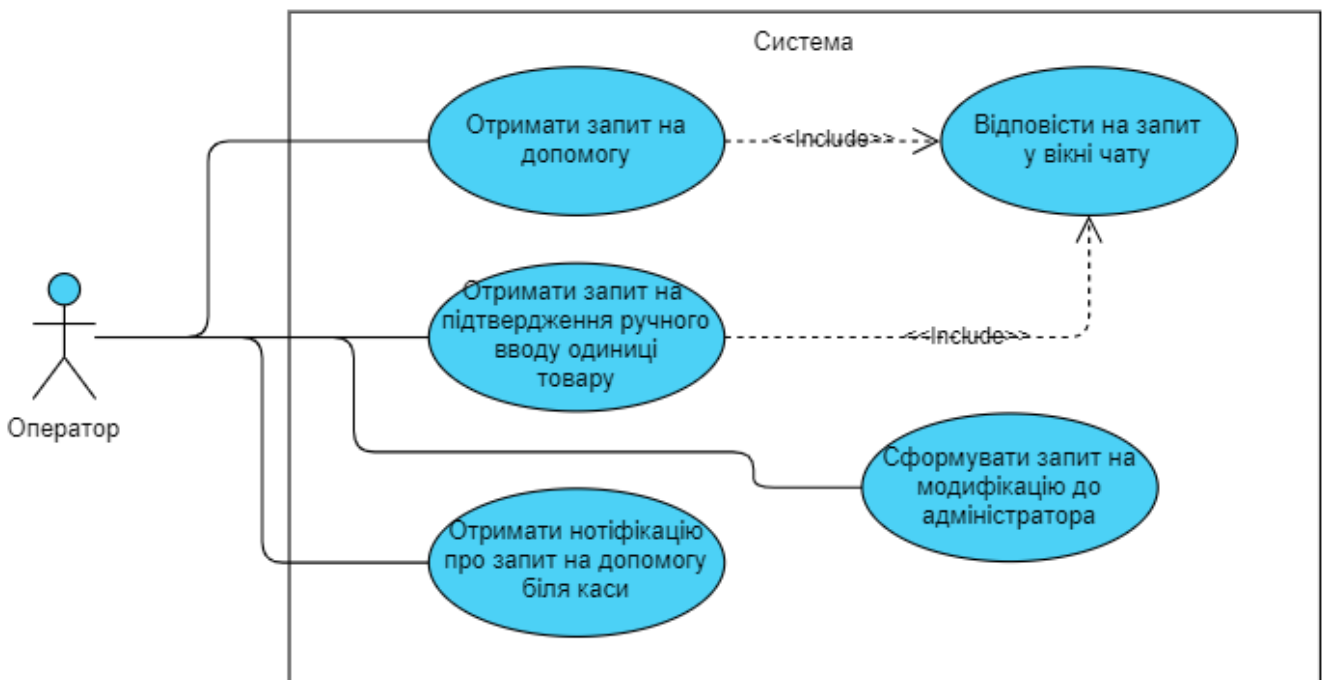


Рисунок 11 – UML діаграма сценаріїв взаємодії оператора зі системою

Оператор системи, окрім звичайних операцій входу систему, які є притаманними звичайному користувачу (покупцю), також може здійснювати наступні дії:

- Отримувати нотіфікацію про запит на допомогу у внутрішньому чаті додатку. В разі отримання подібної нотіфікації, оператор має відкрити чат з користувачем, що надіслав такий запит, та оперативно надати необхідну допомогу, чи відповісти на поставлені запитання.
- Отримувати запит на підтвердження ручного вводу одиниці товару. В разі отримання подібної нотіфікації, оператор має відкрити вікно внутрішнього чату з користувачем, що надіслав даний запит та підтвердити ручне введення коду товару.
- Отримувати нотіфікацію про запит на допомогу біля каси самообслуговування. При отриманні подібного повідомлення, оператор має підійти до вказаної каси та допомогти покупцю з верифікацією, або ж оплатою, кошика.
- Сформувати запит на корекцію товарів для адміністратора

### 3.11. Сценарії взаємодії адміністратора

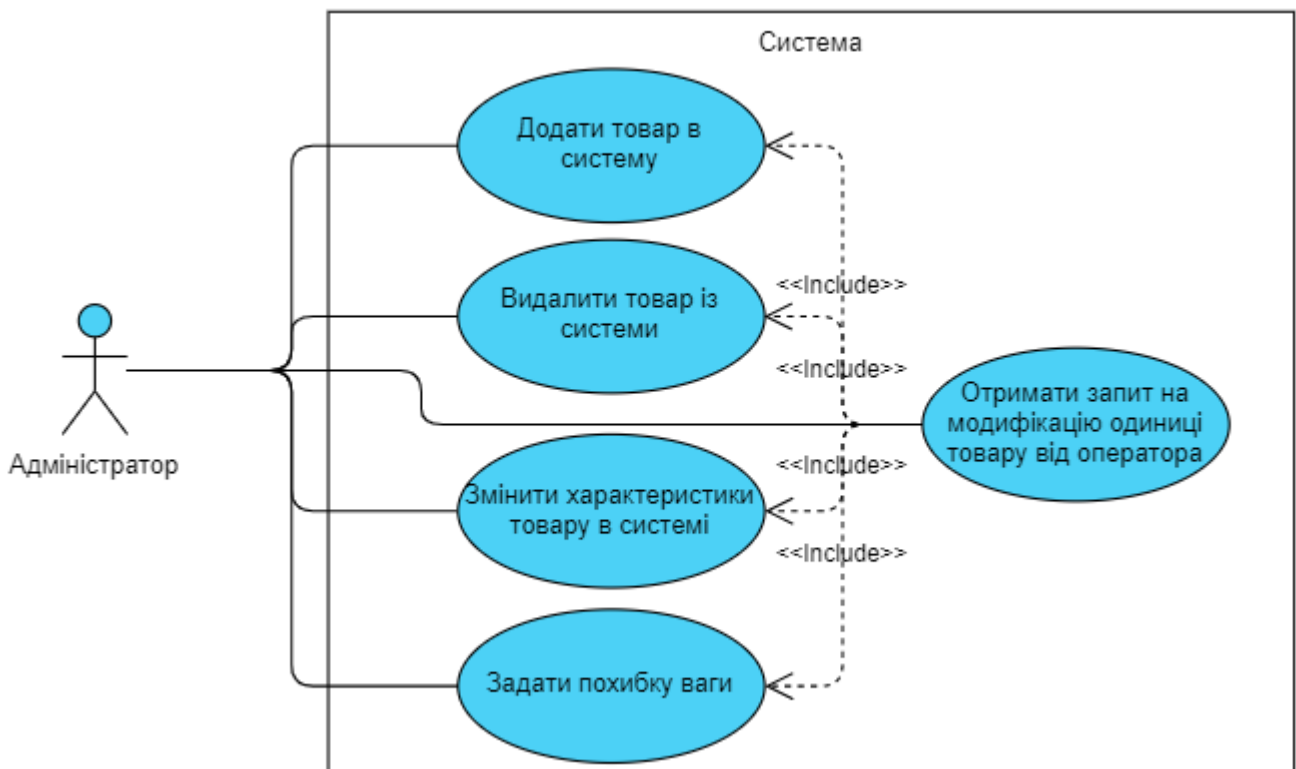


Рисунок 12 – UML діаграма сценаріїв взаємодії зі системою адміністратора

Адміністратор системи, окрім звичайних операцій входу систему, які є притаманними звичайному користувачу (покупцю), також може здійснювати наступні дії:

- Додавати та видаляти товари в системі
- Змінювати характеристики товарів в системі, такі як: кількість, найменування, опис, вага, ціна, і т.п.
- Задавати похибку ваги, що використовується при визначенні допустимого відхилення під час перевірки ваги кошика покупця
- Отримувати запит на модифікацію одиниці товару від оператора. Такий запит може бути сформований в результаті роботи оператора із користувачем системи, який сформував запит на допомогу на тому, чи іншому етапах роботи із системою.

## 4. СТРУКТУРНА СХЕМА СИСТЕМИ

Перш, ніж визначати структурну схему системи, необхідно визначити місце такої системи в загальній картині сфер, в яких працює роздрібна торгівля.

### 4.1. Сфери роботи роздрібно́ї торгівлі

Управління роздрібною торгівлею це складний та багатоетапний процес, який забезпечує контроль усіх бізнес-процесів, операцій та запитів, та допомагає клієнтам придбати бажану продукцію (товари) у фізичних чи цифрових роздрібних магазинах.

Роздрібна торгівля це комплекс різноманітних областей, які впливають одна на одну.



Рисунок ? – Схема областей, в яких діє роздрібна торгівля

Необхідно розуміти, що навіть, якщо магазин роздрібно́ї торгівлі користується різноманітними інструментами для забезпечення жаданих цілей в кожній із зазначених сфер, успішність чи неуспішність інструменту може вплинути на всю еко систему.

Розглянемо детальніше кожну із сфер роботи магазину роздрібної торгівлі, зазначених на рисунку (?):

- Store Operations/Sales (Операції магазину, продажі)

Ця область включає в себе всі інструменти, системи, та підходи, що використовують саме задля успішного підтримки функціонування роздрібної торгівлі – управління людьми, ланцюжок поставок, макет магазину, касові операції, фізичні запаси, управління основними даними, акції та ціноутворення тощо. Тут потрібно зазначити, що функціонування може бути успішним – замовлення будуть швидко оброблятися, товари будуть вчасно замовляються, інвенторика складів буде в ідеальному стані, проте об'єм продажів може не вражати. Цьому допомагають наступні чинники.

- Store design (Дизайн магазину)

Дизайн набув важливого значення в сучасному світі. Сучасному користувачу вже не достатньо мати доступ до правильно функціонуючої системи, при цьому зовсім не важливо це онлайн чи офлайн система. Щоб продуктом захотіла користуватися велика кількість людей, він має бути естетично приємним та викликати позитивні емоції. Цей факт особливо важливий для роздрібної торгівлі, так як в ній все побудовано навколо покупця.

- Product/Brand Management (Менеджмент бренду та продукту)

Бренд це важлива складова будь-якої успішної торгівлі. Одним з визначень бренду є таке, що збирає воєдину суму почуттів, спогадів, емоцій чи образів, що виникають у персони, коли вона з ним (з брендом) взаємодіє. Інші визначення описує бренд, як сукупність високої якості та гарної репутації свого власника. Як ми бачимо, в обох випадках мова знову ж таки йде про позитивні емоції користувача. Наразі бренд є більш фінансовим визначенням. Наприклад, зубна паста від звичайного виробника та



від виробника з гучним іменем будуть відрізнятися ціною. Виробник з гучним іменем вкладає в нього певну «гарантію якості», за яку користувачу пропонується переплатити. Варто, чи ні, здійснювати таку переплату – це вже окреме питання.

- **Merchandising (Мерчендайзинг)**

Це комплекс заходів у роздрібній торгівлі, метою яких є підвищення попиту на продукцію. Також це можна назвати мистецтвом представлення товару у торговому залі. Подібні заходи дозволяють сформувати у споживачів позитивний імпульс, спрямований не тільки на купівлю, а також на створення іміджевого образу бренду у свідомості покупця.

- **Accounting/Finance (Бухгалтерія/Фінанси)**

Бухгалтерія та управління фінансами це безумовно невід’ємна складова будь-якого бізнесу. Потрібно розуміти, що практично завжди будь-які рішення керівництва бізнесу приймаються за допомогою департаменту фінансів. Тобто, в результаті оцінки економічної вигоди, та чи інша пропозиція може бути прийнята, як прибуткова, або ж відхилена, як збиткова.

- **Marketing (Маркетинг)**

Маркетингова сфера спрямована на досягнення цілей бізнесу, підприємств, установ та організацій шляхом формування попиту та максимального задоволення потреб споживачів. Загалом, призначення маркетингу полягає в «визначенні та задоволенні людських і суспільних потреб». Ціллю маркетингу в роздрібній торгівлі є створення привабливої пропозиції користувачу (покупцю). Будь-які акції, пропозиції та знижки є результатом маркетингових ходів.

- **Human Resources (Людські ресурси)**

Ця сфера пов’язана з управлінням всіма працівниками відділів мережі/окремого магазину роздрібної торгівлі. На неї покладено багато функції, що напряду впливають на якість та швидкість роботи

працівників: найм, звільнення, навчання нових працівників, розв'язання робочих конфліктів, контроль за дотриманням прав працівників, тощо. Також, від роботи цього відділу залежить і імідж всього бізнесу – бізнес, що піклується про своїх працівників, заслуговує на повагу та привертає до себе чималу позитивну увагу, що непрямым чином впливає на його фінансові показники, а також допомагає маркетингу, бренд менеджменту, та бухгалтерії.

- **SCM – Supply Chain Management (Управління ланцюгом поставок)**  
Управління поставками це сукупність процесів, що доставляють продукти ,або послуги, від постачальників, через виробництво чи дистрибуцію безпосередньо кінцевому споживачу. Управління поставками також включає в себе управління логісткою, що особливо критично для великих мереж роздрібної торгівлі. Саме тому це є однією з найважливіших областей роздрібної торгівлі.
- **MIS – Management Information System (Інформаційна система управління)**  
Інформаційні системи управління роздрібною торгівлею надають багато різноманітних функцій компаніям, які ними користуються. Зазвичай комп'ютерна програма або інший автоматизований процес, збирає дані про клієнтів, відстежує інвентаризацію, надає електронні точки продажу і проводить маркетингові дослідження. Наприклад, будь-який магазин одягу, швидше за все, використовує таку систему, щоб зберегти ім'я споживача в базі даних програми лояльності, зателефонувати йому про покупку та визначити, чи придбана ним блакитна сорочка є достатньо популярною серед клієнтів, щоб її дозамовити.  
Така система надає інформацію маркетингову, фінансовому, мерчандайзинговому та іншим відділам для прийняття рішень.

Отже, ми розглянули основні сфери діяльності роздрібної торгівлі. Зрозуміло, що всі зазначені області дуже тісно переплітаються та взаємодіють одна з одною. Також є очевидним, що покращення однієї з областей у прямий чи непрямий спосіб впливає на всі інші.

У випадку розробки системи самостійного сканування, основною областю, яку чекає покращення, є Store operations/Sales (Операції магазину / Продажі).

Впровадження системи самостійного сканування також вплине на наступні сфери:

- Управління брендом

Впровадження системи самостійного сканування демонструє бажання бренду слідувати за сучасними технологіями, що впливає на його вартість

- Мерчандайзинг

В разі впровадження система відкриває нові можливості для представлення товару у торговому залі.

- Фінанси

Подібна система дозволить збільшити прибуток мережі роздрібної торгівлі

- Маркетинг

Система сама по собі є досить гучним маркетинговим ходом. При цьому, інтеграція з системою дозволить розширити маркетингові можливості.

- Управління поставками

Для управління поставок система самостійного сканування дозволяє більше точно та швидко визначати попит на певні товари та вчасно їх поставляти.

- Інформаційна система управління роздрібною торгівлею

Самостійне сканування відкриває доступ до інформації у більш персоналізованій формі. Разом з цим, необхідно зазначити, що і збільшується

відподальність власника подібної системи перед кінцевими споживачами. Необхідне розуміння того, яку персональну інформацію, в яких випадках та як можна використовувати. Також необхідно забезпечити користувачу відмову від будь-яких використань його персональної інформації в цілях отримання прибутку.

Отже, розглянемо схему системи управління, що використовується в магазинах роздрібною торгівлі та підсистеми, з яких вона складається.

#### 4.2. Схема системи управління роздрібною торгівлі

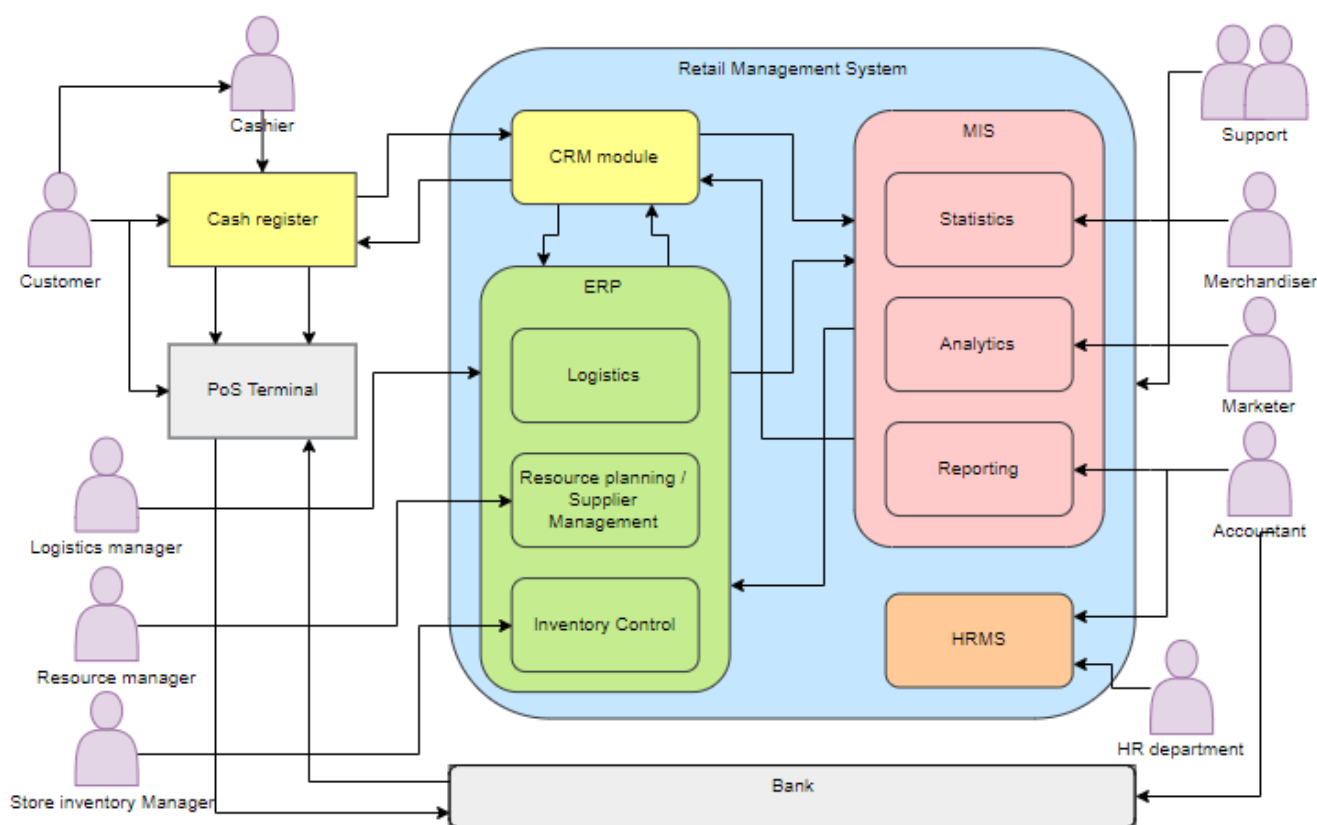


Рисунок ? – структурна схема системи управління роздрібною торгівлі

На рисунку (?) зображена система управління роздрібною торгівлею (Ratil Management System). З малюнку стає зрозумілим, що у цієї системи є кілька користувачів:

- Customer (Клієнт)

В даному випадку, клієнт є кінцевим споживачем.

Клієнт взаємодіє з системою через касу у випадку каси самообслуговування. Також клієнт може взаємодіяти із системою через касира у випадку класичної каси.

Додатково клієнт може взаємодіяти з терміналом оплати, який оброблює банківську транзакцію, у випадку розрахунку банківською картою.

- **Cahier (Касир)**

Касир взаємодіє з системою через касу.

- **Logistics manager (Логістичний менеджер)**

Логістичний менеджер взаємодіє з модулем логістики та займається всіма питаннями, що стосуються доставлення замовлень/закупок/товарів.

- **Resource manager (Ресурсний менеджер)**

Ресурсний менеджер займається питаннями планування закупок, створює задачі для логістики.

- **Store Inventory Manager (Менеджер інвентеризації магазину)**

Такий менеджер займається питаннями наявності товарів та створює запити до ресурсного менеджера з питань замовлення вже розпроданих товарів, чи таких, що закінчуються.

- **Support (Підтримка)**

Служба підтримки може бути як технічною, так і клієнто-орієнтованою, та здійснювати як підтримку кожної з працюючих систем та департаментів, так і кінцевих споживачів.

- **Merchandiser (Мерчандайзер)**

Мерчандайзер взаємодіє зі статистикою, та на основі неї приймає рішення щодо розташування товарів чи їх представлення в онлайн/офлайн магазинах.

- **Marketer (Маркетолог)**

Маркетолог відповідає за маркетингові стратегії, які він будує на основі аналітичної інформації.

- Accountant (Бухгалтер)

Бухгалтер працює зі звітами та фінансовою інформацією. Також для вчасного нарахування заробітної платні, бухгалтер працює з системою управління людськими ресурсами.

- HR department (Департамент людських ресурсів, або ж відділ кадрів)

Відділ кадрів взаємодіє з системою управління людськими ресурсами для найму, звільнення, перекваліфікації і т.д.

Кількість користувачів та різновид задач, з якими ці користувачі працюють, свідчить про комплексність та рівень складності системи. Далі коротко опишемо кожен з елементів та підсистем:

- CRM – Customer Relationship management (Система управління відносинами з клієнтами)

Система управління відносинами з клієнтами надає цілий комплекс інструментів, які використовуються компаніями для управління взаємовідносинами зі споживачами, включаючи збір, зберігання й аналіз інформації про споживачів, постачальників, партнерів та інформації про взаємовідносини з ними.

- ERP – Enterprise resource planning (Система корпоративного планування ресурсів)

Система корпоративного планування ресурсів включає в собі наступні модулі:

- Logistics

Управління логістикою закупок

- Resource planning / Supplier Management

Модуль планування закупок та управління постачальниками

- Inventory control

Контроль наявної кількості товарів

- MIS – Management Information System (Інформаційна система управління)

Складається з наступних модулів:

- Statistics

Статистика продажів

- Analytics

Аналітика продажів та аналіз трендів

- Reporting

Система побудови звітів

- HRMS – Human Resource Management System (Система управління людськими ресурсами)

Система зберігає в собі всю інформацію про працівників, таку як паспортні дані, дані про освіту, договори та інші необхідні для звітності документи.

#### **4.3.Бізнес процес покупки**

В попередніх розділах ми розглянули всі сфери діяльності, в рамках яких функціонує система управління роздрібною торгівлею. Також було розглянуто структурну схему такої комплексної системи. Дана робота пропонує ввести додаткову систему самостійного сканування. Для того, щоб зрозуміти, яку проблему ця система пропонує вирішити, розглянемо бізнес процес покупки.

З Рисунка (?) бачимо, що процес покупки починається з вибору товарів (1), після чого настає фаза завершення покупки (2). Завершення покупки (2) може відбуватися двома шляхами: через касира (3), якщо це класична каса, або ж самостійно, якщо це каса самообслуговування.

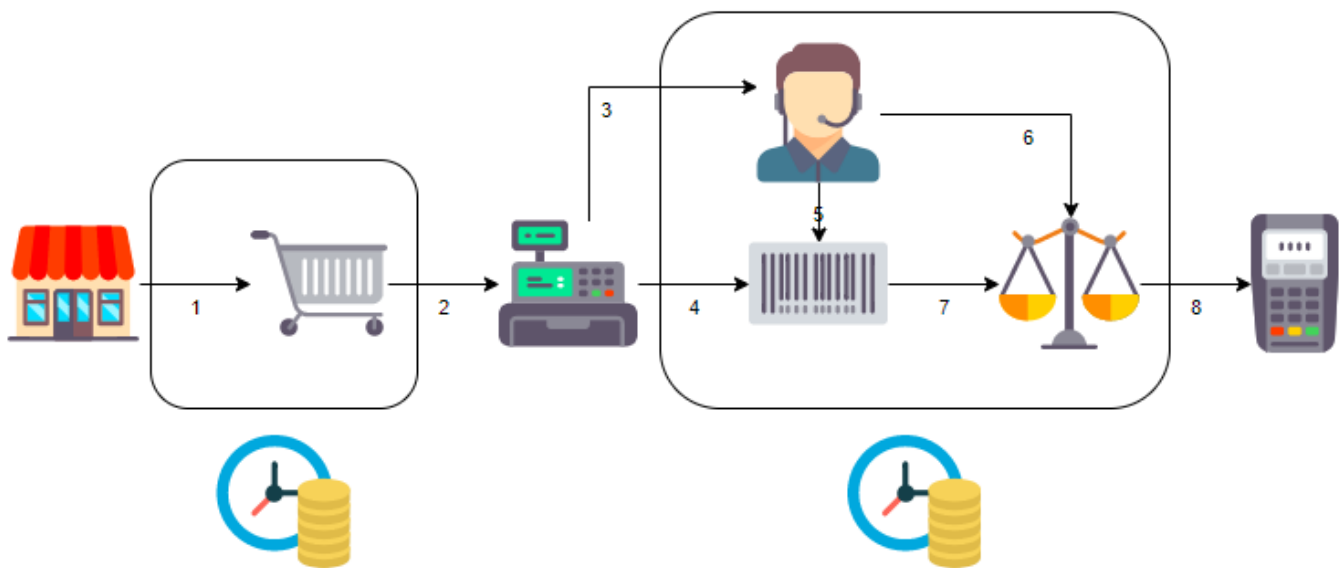


Рисунок (?) – Бізнес процес створення покупки

В разі класичної каси, касир сканує (5) кожен товар та зважує його (6). У випадку каси самообслуговування, клієнт самостійно сканує кожен товар (4) та кладе його на ваги (7).

В самому кінці відбувається оплата покупки.

Як бачимо з малюнку, в описаному процесі є 2 операції, на які клієнт витрачає більше всього часу – безпосередньо сам процес вибору товарів, та сканування при завершенні замовлення. Якщо процес вибору товарів неможливо оптимізувати, так як цей процес повністю контролюється та залежить від клієнту, то процес завершення покупки піддається оптимізації.

Розглянемо, як виглядатиме такий само процес покупки за використання системи самостійного сканування (Рисунок ?).

При використанні системи самостійного сканування клієнту пропонується самостійно сканувати товари (2) за допомогою смартфона в процесі їх вибору та наповнення корзини (1). Процес завершення покупки (3) в цьому випадку виглядає дуже просто – користувачеві достатньо одноразово просканувати на касі QR-код своєї покупки (4) та зважити товари (5). Завершальним кроком є оплата товару (6).



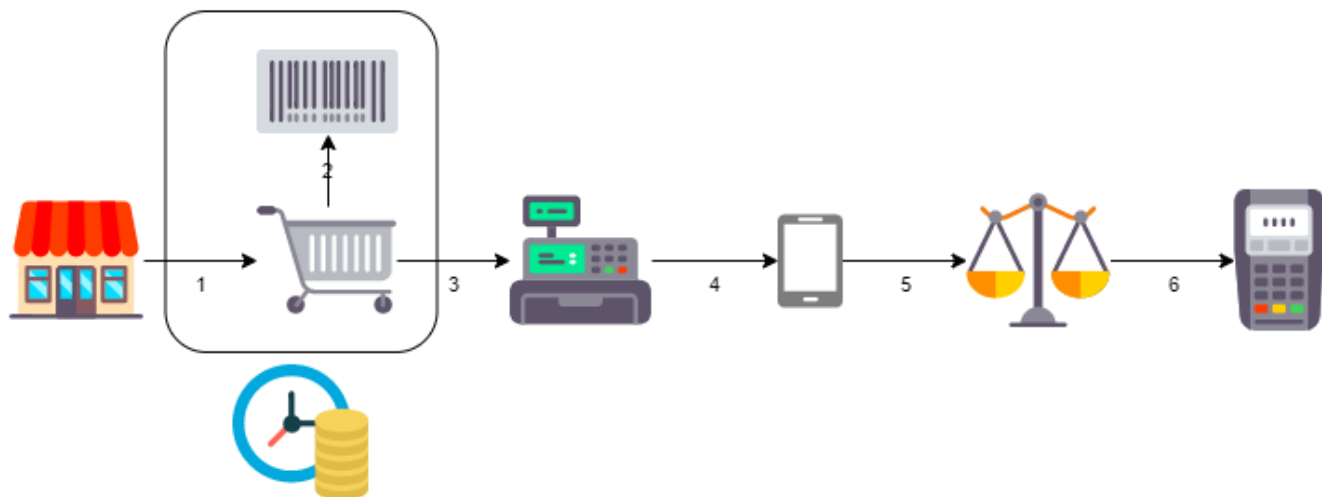


Рисунок ? – Бізнес процес створення покупки із самостійним скануванням

Як видно з малюнку (?), система самостійного сканування дозволяє максимально можливо скоротити час перебування клієнта в магазині.

#### 4.4. Структурна схема системи самостійного сканування

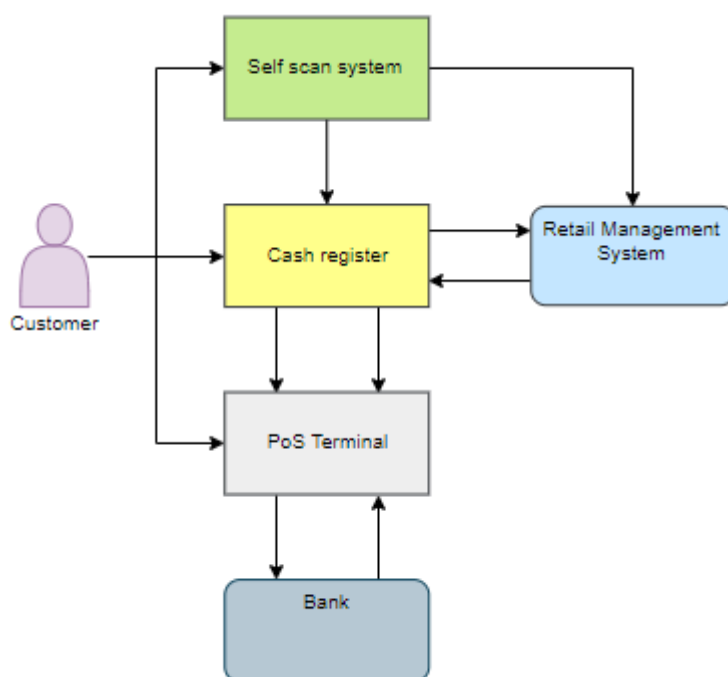


Рисунок (?) – Структурна схема системи самостійного сканування

Система самостійного сканування являє собою прошарок між клієнтом та системою управління роздрібною торгівлею.

За використання системи самостійного сканування, клієнт взаємодіє з системою управління роздрібною торгівлею через 3 інтерфейси:

- Мобільний додаток, який дозволяє сканувати товари, та тим самим наповнювати кошик.
- Касовий апарат, котрий сканує QR-код з мобільного додатку, та передає всю необхідну інформацію до терміналу оплати (PoS terminal).
- PoS terminal (Термінал оплати), який оперує банківською транзакцією користувача та дозволяє оплатити кошик товарів.

З Рисунок (?) ми також бачимо, що система самостійного сканування надає інформацію системі управління роздрібною торгівлею. Загалом, ця взаємодія може бути двух-сторонньою та дуже сильно залежить від імплементації конкретної інтеграції між двома визначеними системами. Таких інтеграцій може бути кілька, та вони можуть бути специфічні для кожного окремого замовника подібної системи самостійного сканування. Саме через неможливість створення уніфікованого інтеграційного інтерфейсу, інтеграції з зовнішніми системами заплановані як найступний крок розвитку даної роботи.

## **5. ВИБІР ТА ОБГРУНТУВАННЯ ЕЛЕМЕНТІВ ТА ТЕХНОЛОГІЙ**

### **5.1. Вибір архітектурного підходу**

Архітектура системи це сукупність елементів, що мають певну форму (властивості та накладені обмеження), та їх обґрунтування. Обґрунтування фіксує мотиви вибору певного архітектурного стилю, елементів та обмежень.

Проектування архітектори системи ставить за задачу перетворення вимог до системи у вимоги до програмного забезпечення, та в кінцевій точці створення архітектури такого програмного забезпечення, що буде задовільняти вказаним вимогам. Зазвичай, саме нефункціональні вимоги впливають на будову системи, проте і функціональні можуть мати значний вплив на її кінцевий вигляд.

Побудова архітектури системи здійснюється шляхом визначення цілей системи, її вхідних і вихідних даних, декомпозиції системи на підсистеми, компоненти або модулі та розроблення її загальної структури. Проектування архітектури системи може проводитися різними методами (стандартизованим, об'єктно-орієнтованим, компонентним і ін.), кожний з яких пропонує свій шлях побудови архітектури, а саме, визначення концептуальної, об'єктної й інших моделей за допомогою відповідних конструктивних елементів (блок-схем, графів, структурних діаграм тощо).

Існує кілька архітектурних стилей:

#### **1. Client-server architecture**

Клієнт-серверна архітектура є однією із самих популярних типів архітектури програмного забезпечення, яку обирають для побудови систем в сучасному світі. В її основі лежить створення розподіленої та віддаленої системи, де елементи діляться на клієнтів, які користуються сервісами, та сервери, що надають ці сервіси. Важливим моментом такої архітектури є мережа, що забезпечує комунікацію між клієнтами та серверами.

#### **2. Component-based architecture**

Компонентно-орієнтована архітектура є різновидом клієнт-серверної архітектури з тією відмінністю, що серверна частина представляється у вигляді

набору компонент, що надають одна одній сервіси та відповідають на окремі набори функцій та операцій. Така архітектура дозволяє інкапсулювати бізнес логіку та створити прозорі та логічні компоненти, кожна з яких займатиметься своєю частиною обробки інформації. Потрібно зазначити, що компоненти також можуть бути окремими прикладеннями.

### 3. Service-oriented architecture

Сервіс-орієнтована архітектура базується на компонентах, що надають сервіси для інших компонент системи використовуючи комунікаційні протоколи мережі.

### 4. Microservices

Мікросервісна архітектура це різновид сервіс-орієнтованої архітектури, що складається зі слабозав'язаних один на одного сервісів. Сервіси в такій архітектурі добре гранульовані та незалежні, а протоколи взаємодії легко-важні, задля збільшення швидкості обробки інформації.

Під час визначення підходу до побудови архітектури системи також потрібно мати на увазі, що в сучасному світі системи мають бути розподіленими. А з цього витякають наступні принципи, які переслідує розподілена архітектура:

#### 1) Забезпечення з'єднання

Користувачу має надаватися легкий доступ до ресурсів системи

#### 2) Безпека з'єднання

Будь-які чутливі дані (паролі, логіни та інша персональна інформація) не мають зберігатися чи передаватися у відкритому вигляді.

Одним із самих популярних засобів це шифрування за допомогою SHA256. Особисті дані клієнту шифуються на стороні клієнту і лише тоді передаються на сторону серверу.

#### 3) Прозорість

Розподільна система має проявляти свою прозорість у наступних аспектах:

- Доступ до ресурсів – приховування різниці в поданні даних та в доступі до них
- Розташування – приховування місця розташування ресурсу (для користувача має бути відсутньою різниця у користуванні тим чи іншим ресурсом не залежно від його фізичного місцезнаходження)
- Перепідключення та зміна розташування – приховування переміщення ресурсу у процесі обробки запиту користувача
- Реплікація – для користувача має бути неочевидним той факт, що під час його користування системою, ресурси, що надають йому інформацію взаємо замінилися (наприклад, при користуванні системою, користувач мав доступ до ресурсу, що підтримувався обчислювальною машиною; у разі її виходу з ладу користувач не має відчутти заміни обслуговування реплікацією)
- Одночасний доступ – приховування факту доступу кількох користувачів до одного й того самого ресурсу одночасно. При цьому, система має тримати даними консистентними за умови редагування одного й того ж ресурсу різними користувачами одночасно.
- Відмова – приховування виходу з ладу обчислювального ресурсу. Тип не менш, не завжди цей момент можна приховати, так як вихід з ладу одного з обчислювальних ресурсів може вплинути на швидкодію усієї системи в цілому.
- Оперування даними – приховування факту того, звідки користувач отримав дані – з оперативної системи, з файлової чи з бази даних.

#### 4) Масштабованість

Висока масштабованість системи є запорукою успішного функціонування системи під час її розширення чи навпаки звуження. Тобто, додавання нового серверу прикладення, або ж додаткової ноди бази даних має проходити безболісно для працюючої системи.

#### 5) Відмовостійкість

Висока відмовостійкість системи дозволяє спокійно витримувати високі навантаження, продовжувати роботу під час вікон технічного обслуговування, чи вистояти неполадки одного або кількох вузлів системи.

Оглянувши кілька архітектурних стилей та вимоги до розподілених систем, можна зробити висновок, що класична клієнт-серверна монолітна система в даному випадку зовсім не підходить під описані цілі та визначені попередньо вимоги. При цьому, мікросервісна архітектура хоча і виглядає привабливою, проте поділить систему самостійного сканування на такі дрібні сервіси, що такий поділ не буде доцільним та вимагатиме багато ресурсів (процесорів, пам'яті). Перехід на мікросервісну архітектуру є можливим та виправданим за умови росту функціоналу системи та її масштабів охоплення. Для прикладу, якщо система самостійного сканування в майбутньому прийме на себе частину функціоналу CRM системи, буде доцільним виділення окремих незалежних сервісів в її рамках.

На даний момент єдиним вірним рішення є поєднання клієнт-серверної моделі з сервіс-орієнтований підходом.

Переваги SOA (Сервіс-орієнтованої архітектури):

- 1) Скорочення часу реалізації проектів, або «часу виходу на ринок».
- 2) Підвищення продуктивності.
- 3) Більш швидка і менш дорога інтеграція додатків і інтеграція B2B — зупинимося більш детально на цьому пункті.
- 4) Простіша розробка і впровадження додатків.
- 5) Використання поточних інвестицій.
- 6) Зменшення ризику, пов'язаного з впровадженням проектів в області автоматизації послуг і процесів.
- 7) Можливість безперервного поліпшення наданої послуги.
- 8) Скорочення числа звернень за технічною підтримкою.
- 9) Підвищення показника повернення інвестицій (ROI).

- 10) Перспективи подільшого розвитку
- 11) Досить простий перехід на модель мікросервісної архітектури за такої необхідності

Сервіс орієнтована архітектура має всі переваги, вирішує поставлені задачі та відповідає визначеним вимогам. Наступним кроком розглянемо технології, що використовуються для її реалізації.

## **5.2. Вибір технологій сервісно-орієнтованої архітектури**

Основні технології сервісно-орієнтованої архітектури, що використовуються в сучасних системах:

### **1) RPC (Remote Procedure Call)**

Технологія між-процесного обміну даних, що дозволяє викликати функцію або здійснити передачу даних іншому процесу, що виконується локально або ж віддалено. Необхідна умова – наявність єдиної мережі.

### **2) SOAP (Simple Object Access Protocol)**

Стандартизований протокол, що дозволяє реалізувати обмін структурованою інформацією через веб-сервіс. Здебільшого інформація передається в форматі XML.

### **3) REST (Representational State Transfer)**

Підхід до побудови обміну структурованою інформацією через веб-сервіси. Працює на протоколі HTTP. Є більш популярним за SOAP саме через більш широку застосованість протоколу та простоту його використання. Здебільшого інформація передається в форматі JSON.

Програмний інтерфейс:

#### **1) WEB-based**

Інтерфейс системи доступний через вебсайт. Є універсальним способом, так як не залежить від операційної системи, та її конфігурації, кінцевого користувача.

#### **2) Desktop-based**

Інтерфейс системи доступний через прикладення на персональному комп'ютері. Є чутливим до операційної системи комп'ютера, на якому його запускають.

### 3) Mobile-based

Інтерфейс системи доступний через прикладення на смартфоні. Також є чутливим до операційної системи. Проте, в даному випадку це не є критичним моментом з урахунком того, що операційною системою Android користується більш ніж 74% користувачів мобільного ринку. Тоді, як на iOS припадає всього 22%.

ORM-фреймворки, що реалізують JPA (Java Persistence API)

#### 1) Hibernate

Технологія надає інструменти для комунікації з базою даних. Це самостійний фреймворк, що знаходиться у вільному доступі.

#### 2) Spring Data JPA

Також є технологією, що надає інструменти для доступу до бази даних. Поставляється в комплексі з платформою Spring, яка шикоро використовується в різноманітних сферах та надає безліч корисних інструментів для побудови програмного забезпечення.

Web-фреймворки рівня представлення даних для JAVA:

#### 1) GWT

#### 2) JavaFX

#### 3) HTML/CSS/JavaScript

З перелічених технологій та підходів виберемо ті, що найкраще підходять для вирішення поставленої задачі: клієнт-серверна модель, сервісно-орієнтована архітектура, REST веб-сервіси, Web-based та Mobile-based програмні інтерфейси, Android, Spring та HTML/CSS/Javascript.



### 5.3. Обґрунтування обраних технологій

#### *Клієнт-серверна модель*

Вимогами до системи, що розробляється, чітко визначено роль клієнту та об'єм функціоналу, що йому надається, що найкраще відповідає клієнт-серверній моделі. Додатковим плюсом за такою моделлю є переніс всього «важкого» функціоналу, як взаємодію з базою даних, на сторону серверу. Це дозволяє спростити сторону клієнту та надавати більш швидкий доступ до оброблених даних за рахунок серверних ресурсів та обчислювальних можливостей.

#### *Сервісно-орієнтована архітектура*

Забезпечує слабку зв'язність сервісів, що сприяє їх максимальній ізоляції. Це дозволяє більш гнучко будувати систему, а також в разі необхідності змінювати програмний код чи бізнес-логіку, правила взаємодії, протокол взаємодії, тощо. окремого сервісу, а не всієї системи в цілому.

#### *REST веб-сервіси*

Підхід, що широко використовуються практично в кожній розподіленій системі. Спрощує інтеграцію з іншими системами, надає гнучкі інструменти налаштування безпеки. Безумовною перевагою є робота через HTTP/HTTPS протоколи.

#### *Web-based інтерфейс*

Інтерфейс дозволяє не залежити від операційної системи персонального пристрою, через який здійснюється доступ до системи

#### *Mobile-based інтерфейс*

Відповідає поставленим вимогам, за яких кінцевий користувач системи має використовувати свій смартфон в якості сканеру, та з його допомогою наповнювати кошик.

#### *Android платформа*

Вибір цієї платформи зумовлений тим, що більше 74% користувачів смартфонів, користуються такими, що використовують Android в якості операційної системи.

### *Spring фреймворк*

Фреймворк надає багато різноманітних інструментів, що дозволяють будувати веб-сервіси, а також рівні доступу до даних в уніфікованому вигляді. Використання Spring фреймворку дозволяє отримати всі необхідні інструменти в одному місці та відмовитись від використання бібліотек, що не створюють єдину еко-систему, а відповідно і від затрат на написання коду, який би поєднував їх воєдине.

### *HTML/CSS/JavaScript*

Інструменти дозволяють розроблювати гнучкий та простий UI, який буде підтримуватись всіма веб-браузерами (програмними комплексами для відображення веб-сторінок)

## **5.4. Вимоги до бази даних**

Наразі існує велика кількість різноманітних баз даних, які надають функціонал для зберігання даних. Бази даних не тільки визначають спосіб зберігання даних, але і доступу до них.

Визначимо основні вимоги до бази даних, що буде задовольняти вимогам до системи самостійного сканування:

1. База даних має зберігати великі об'єми даних
2. База даних повинна мати мізерний час відклику
3. База даних має вміти працювати з індексами
4. База даних має вміти працювати з реляційною моделлю даних
5. База даних має бути такою, що підтримує транзакції
6. База даних має підтримувати основні принципи ACID:

- Atomicity (Атомарність)

Атомарність гарантує виконання комплексу операцій над даними в рамках однієї транзакції.

- Consistency (Консистентність)

База даних має перебувати в узгодженому стані. Завершення транзакції переводить базу з одного узгодженого стану в інший. При цьому, база

може перебувати в неузгодженому стані під час виконання транзакції, що приводить нас до принципу ізоляції.

- Isolation (Ізоляція)

База даних забезпечує невидимість неузгодженого стану (проміжних змін) за рамками транзакції. Тобто, якщо одна транзакція почала виконувати дії над таблицею, інша транзакція не зможе ці зміни побачити, поки перша не буде завершеною.

- Durability (Живучість або Довговічність)

База даних забезпечує збереження результатів транзакції в умовах нестабільності системи. Що значить, що в разі успішного проведення транзакції, після відновлення з аварійного стану, база даних має зберегти результати цієї транзакції.

7. База даних має підтримувати реплікацію даних або шардування.

## 5.5. Обґрунтування вибору бази даних

Наведемо порівняльну характеристику кількох баз даних (Таблиця ?)

	Oracle	PostgreSQL	MySQL	MongoDB	Cassandra
SQL син-таксис	1	1	1	0.5	0
Об'єм даних	1	1	0	0	1
Швидкість читання	1	1	1	0.5	0
Швидкість запису	0.5	0.5	0.5	1	1
Реляційна модель	1	1	1	0	0

Транзакційність	1	1	1	0.5	1
Атомарність	1	1	1	1	1
Консистентність	1	1	1	0.5	1
Ізоляція	1	1	1	0.5	0.5
Довговічність	1	1	1	1	1
Реплікація	0	0.5	0	1	1
Безкоштовна ліцензія	0	1	1	1	1
Підсумок	9.5	11	9.5	7.5	8.5

Базуючись на порівняльній характеристиці, очевидним та самим задовільним рішенням буде база даних PostgreSQL. Безумовними перевагами цієї бази є підтримка реляційної моделі даних, підтримка ACID принципів, наявність індексів та підтримка SQL синтаксису, а також безкоштовної ліцензії, що є безумовною перевагою. Слабким місцем бази є відсутність повноцінної репліки, проте це вирішується шардуванням бази.

## 6. ER-ДІАГРАМА

### 6.1. Модель бази даних

Для системи самостійного сканування було обрано реляційну модель бази даних.

Реляційна модель даних визначається як логічна модель даних. Реляційна модель даних забезпечує високий рівень абстракції даних за рахунок того, що така модель надає засоби та інструменти опису даних без необхідності введення додаткових структур з ціллю зробити таку модель більш зручною для комп'ютерних систем.

Основною теорією реляційної моделі даних є теорія нормалізації. У такої моделі визначають наступні складові:

- Структурна

Всі відносини представляються у формі таблиць, де кожен рядок є кортежем, а стовпчик – атрибутом, що визначений в домені. Така форма дозволяє представляти базу даних у вигляді сукупності таблиць.

- Маніпуляційна

Визначає реляційну алгебру та реляційне числення як єдині форми маніпулювання даними

- Цілісна

Складається з двох принципів:

1) Будь-який рядок таблиці відмінний від іншого рядка цієї ж таблиці. Тобто, таблиця має мати первинний ключ, який визначає унікальність рядка в ній.

2) За наявності зовнішнього ключа, для будь-якого рядка таблиці, має знайтися відповідний рядок з таблиці, на яку зовнішній ключ посилається. Також, значення такого ключа може бути невизначеним.

Для баз даних, що підтримують реляційну модель даних властиво підтримувати також транзип транзакційності, а також основні принципи ACID (Атомарність, Консистентність, Ізоляція та Довговічність)

## 6.2. Схема бази даних

Дотримуючись реляційної моделі даних, було розроблено схему бази даних, зображену на рисунку (?).

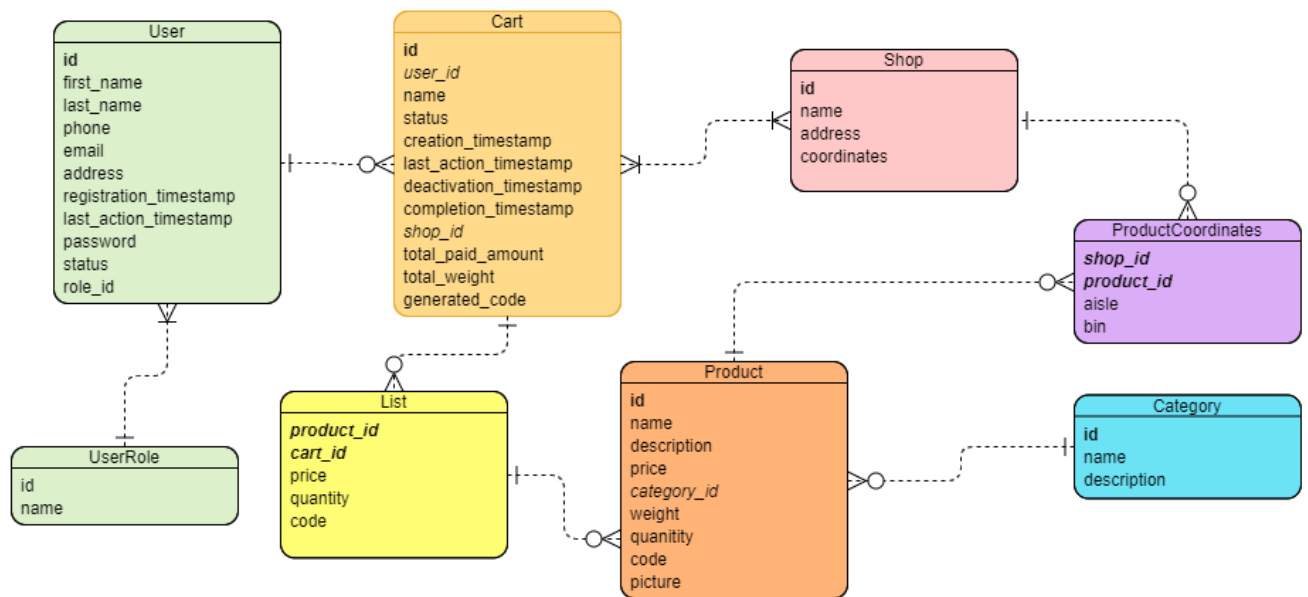


Рисунок ? – Схема бази даних

Розглянемо детально кожну з таблиць та їх поля.

### User

Таблиця зберігає в собі всю інформацію про кінцевого користувача

Таблиця ? – Опис таблиці User

Ім'я колонки	Тип	Опис
id	UUID	Унікальний ідентифікатор користувача
first_name	TEXT	Ім'я користувача
last_name	TEXT	Прізвище користувача

phone	INT	Номер телефону користувача
email	TEXT	Адреса електронної скриньки
address	TEXT	Поштова адреса користувача
registration_timestamp	TIMESTAMP	Дата та час реєстрації користувача
last_action_timestamp	TIMESTAMP	Дата та час останньої активності користувача
password	TEXT	Пароль користувача в зашифрованому вигляді
status	TEXT	Статус аккаунта користувача. Доступні значення: Registered, Active, Removed
role_id	UUID	Ідентифікатор ролі користувача, зовнішній ключ

## UserRole

Таблиця містить ролі користувачів, створені в системі.

*Таблиця ? – Опис таблиці UserRole*

Ім'я колонки	Тип	Опис
id	UUID	Унікальний ідентифікатор ролі користувача
name	TEXT	Назва ролі користувача

## Cart

Таблиця містить в собі кошики, які користувачі створюють в системі при початку процесу покупки.

Таблиця ? – Опис таблиці Cart

Ім'я колонки	Тип	Опис
id	UUID	Унікальний ідентифікатор кошика користувача
name	TEXT	Ім'я кошика користувача
user_id	UUID	Унікальний ідентифікатор користувача, якому належить даний кошик. Зовнішній ключ
status	TEXT	Статус кошика, може бути: Active, Cancelled
creation_timestamp	TIMESTAMP	Дата та час створення кошика користувачем
last_action_timestamp	TIMESTAMP	Дата та час останньої дії в кошику користувачем
deactivation_timestamp	TIMESTAMP	Дата та час деактивації кошика користувачем
completion_timestamp	TIMESTAMP	Дата та час завершення кошика користувачем
shop_id	UUID	Унікальний ідентифікатор магазину, в якому користувач робить покупку
total_paid_amount	NUMERIC(10,2)	Загальна сума кошику, яка була заплачена
total_weight	NUMERIC(10,2)	Загальна вага кошику



generated_code	TEXT	Згенерований системою код при завершенні покупки користувачем
----------------	------	---

## Product

Таблиця зберігає в собі список товарних одиниць, які доступні до покупки користувачем. Саме ця таблиця відображає актуальну ціну товарної одиниці та її наявну кількість.

Таблиця ? – Опис таблиці Product

Ім'я колонки	Тип	Опис
id	UUID	Унікальний ідентифікатор товарної одиниці
name	TEXT	Найменування товарної одиниці
description	TEXT	Детальний опис товарної одиниці
price	NUMERIC(10,2)	Ціна товарної одиниці
category_id	UUID	Унікальний ідентифікатор категорії, до якої належить товарна одиниця
weight	NUMERIC(10,2)	Вага товарної одиниці
quantity	INT	Кількість товарних одиниць
code	TEXT	Код товарної одиниці
picture	TEXT	Шлях до картинки на сервері прикладення, що зображає товарну одиницю

## List

Таблиця допомагає сформувати список товарних одиниць, які належать до одного й того самого кошика.

Таблиця ? – Опис таблиці List

Ім'я колонки	Тип	Опис
product_id	UUID	Унікальний ідентифікатор товарної одиниці, що була додана до кошику. Зовнішній ключ
cart_id	UUID	Унікальний ідентифікатор кошику, до якого була додана товарна одиниця. Зовнішній ключ.
price	NUMERIC(10,2)	Ціна товарної одиниці, що була додана до кошику на момент додавання
quantity	INT	Кількість одиниць однієї й тої самої товарної одиниці, доданої до кошику
code	TEXT	Код товарної одиниці на момент її додавання до кошику

## Shop

Таблиця містить в собі інформацію про магазин роздрібної торгівлі

Таблиця ? – Опис таблиці Shop

Ім'я колонки	Тип	Опис
--------------	-----	------

id	UUID	Унікальний ідентифікатор магазину роздрібної торгівлі
name	TEXT	Найменування магазину роздрібної торгівлі
address	TEXT	Адреса магазину роздрібної торгівлі
coordinates	TEXT	Точні географічні координати магазину роздрібної торгівлі

### Category

Таблиця містить в собі всі доступні категорії товарів

Таблиця ? – Опис таблиці Category

Ім'я колонки	Тип	Опис
id	UUID	Унікальний ідентифікатор категорії товарних одиниць
name	TEXT	Назва категорії
description	TEXT	Детальний опис категорії товарних одиниць

### ProductCoordinates

Таблиця містить в собі відомості про розташування товару всередині магазину роздрібної торгівлі.

Таблиця ? – Опис таблиці ProductCoordinates

Ім'я колонки	Тип	Опис
shop_id	UUID	Унікальний ідентифікатор магазину роздрібної торгівлі. Зовнішній ключ
product_id	UUID	Унікальний ідентифікатор товарної одиниці
aisle	INT	Номер проходу, в якому знаходяться товарні одиниці
bin	INT	Номер секції, в якій знаходяться товарні одиниці

## **7. РЕАЛІЗАЦІЯ БІЗНЕС-ЛОГІКИ СИСТЕМИ**

### **7.1. Архітектура програмного продукту**

На основі розглянутої попередньо інформації, визначимо загальну архітектуру програмного продукту та його основні рівні:

- Рівень бази даних (Database layer) - Повністю забезпечується базою даних
- Рівень доступу до даних (Data Access layer) - Реалізує доступ до бази даних – відкриття з'єднання, створення транзакції, виконання запитів в базі даних, закриття транзакції та інші.
- Рівень бізнес-логіки (Business logic layer) - Реалізує перетворення інформації, введеної користувачем, в інформацію, яка є зрозумілою для системи.
- Рівень сервісів (Service layer) - Реалізує інтерфейс, через який проходять всі запити в систему
- Рівень представлення даних (Presentation layer) - Реалізує рівень, що відповідає за візуальне зображення веб сторінок

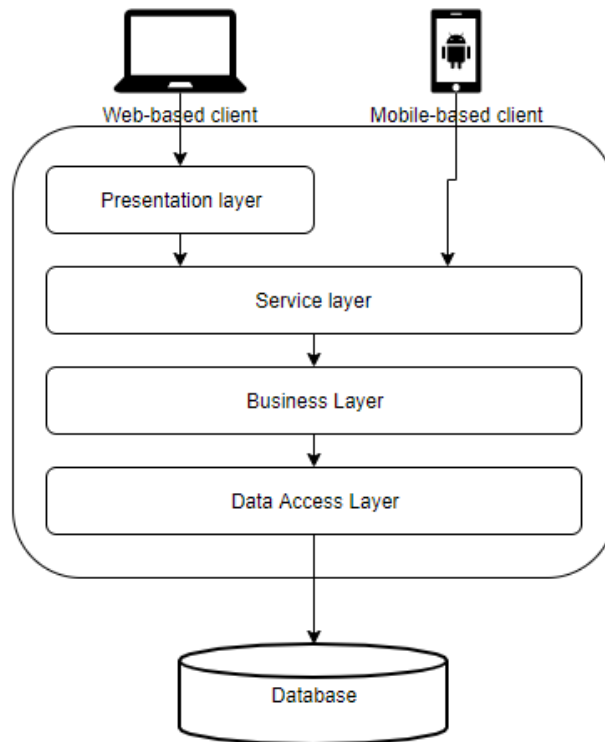


Рисунок ? – Загальна архітектура системи

Як бачимо з рисунку ?, у системи є два клієнти. Одним виступає веб-браузер користувача, іншим мобільне прикладення на смартфоні користувача. У першому випадку під час користування веб-сайтом запити відправляються до серверної частини (Application Server). Кожен такий запит ініціює відповідний виклик на рівні представлення даних.

Слідом за цим, в процес включається сервісний рівень, який точно визначає що за дані прийшли до системи, наскільки вони вірні, і які подальші дії необхідно над ними виконувати. Сервісний рівень являє собою прошарок між візуальним представленням даних користувачеві та бізнес-логікою, що реально виконується на серверній стороні, та надає зручний до неї доступ.

Перетвореннями та маніпуляціями даних займається рівень бізнес логіки. Саме він вирішує в якому вигляді та в якій таблиці будуть відображені зміни, ініційовані користувачем. Наступним кроком є з'єднання з базою даних та запис необхідних змін. В разі, якщо операція, яку запрошував користувач, пройшла успішно, веб-сторінка відображає очікуваний результат. У випадку виникнення помилки, рівень

презентації даних визначає рівень помилки і в залежності від цього інформує користувача належним чином.

У випадку мобільного клієнта рівень презентації даних не приймає участі в описаному ланцюгу, так як він реалізований не на серверній, а на клієнтській стороні. Всі запити, які формуються в результаті дій користувача в мобільному прикладенні, відразу надходять до сервісного рівня.

Зобразимо детальну архітектуру програмного продукту:



Рисунок ? – Загальна архітектура програмного продукту

Детально розглянемо кожен з рівней.

### 7.1.1. Рівень даних та доступу до них

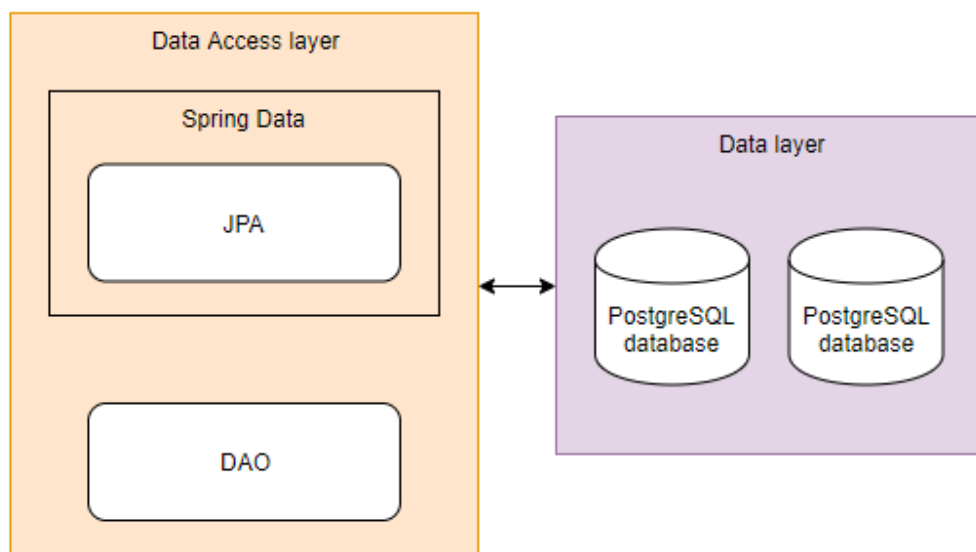


Рисунок ? – Архітектура рівня даних та доступу до них

З рисунка ? зрозуміло, що рівень даних представляє собою база даних PostgreSQL, яка зберігає в собі всю інформацію щодо користувачів, корзин, продуктів, магазинів, тощо.

Взаємодію рівня доступу даних з базою даних PostgreSQL забезпечує JDBC-драйвер, який є платформо незалежним промисловим стандартом взаємодії програмних продуктів, написаних на Java, з різноманітними СУБД.

Кожна з баз даних реалізує свій JDBC-драйвер, який відповідає за наступний функціонал з клієнтської сторони:

- Створення сесії
- Закриття сесії
- Відкриття транзакції
- Закриття транзакції шляхом commit чи rollback операцій
- Виконання SQL запиту
- Виконання SQL запиту, що повертає результат
- Виконання групи SQL запитів

Фреймворк Spring Data JPA дозволяє трансформувати дані з СУБД в об'єкти, якими звично управляти в програмному коді Java, та навпаки, трансформує об'єкти в дані, що зберігаються у базі даних. Такі об'єкти називаються POJO-об'єктами, саме з ними працює рівень DAO. Цей рівень розгортається на сервері додатків.

DAO – це популярний підхід відокремлення доступу до бази даних в окремий рівень, який дозволяє ізолювати всі операції з базою даних в одному місці та мінімізувати витрати, що супроводжують перехід програмного продукту на іншу базу даних, або ж розширення існуючого функціоналу. Такий підхід також дозволяє краще гранулювати задачі для розробників та добиватися результатів в коротші терміни.

Також, DAO рівень має підтримувати логічне шардування бази даних за такої необхідності. Шардування бази даних – це розподіл даних між двома інстансами бази даних. Шардування може бути також логічним, тобто таким, за якого саме прикладний рівень вирішує в якій саме базі зберігається та, чи інша інформація.



### 7.1.2. Рівень бізнес логіки



Рисунок ? – Архітектура рівня бізнес-логіки

Рівень бізнес логіки тісно взаємодіє з рівнем доступу до даних. Будь-яка операція, яка в результаті має відобразити трансформацію даних, має своїм кінцевим пунктом призначення звернення до рівня доступу даних задля збереження нових даних, або ж для читання вже записаних попередньо.

За бізнес логіку відповідають наступні компоненти:

- User Manager

Компонента відповідає за весь функціонал та зміни, що відбуваються з користувачем та його аккаунтом. Зміна паролю, створення нового користувача, зміна електронної адреси і т.п. – область відповідальності компоненти управління користувацькими записами.

- Cart Manager

Компонента відповідає за весь функціонал та зміни, що відбуваються з кошиком користувача. Через цю компоненту проходить процес створення кошику, процес додавання нового продукту до кошику і т.д.

- **Product Manager**

Компонента надає функціонал для роботи з товарними одиницями. Вона займається реєстрацією нового товару в системі, створенні для нього QR-коду, зміною кількості товарних одиниць в магазині роздрібної торгівлі, зміні роштаксування товарних одиниць в магазині роздрібної торгівлі і т.д.

- **Shopping List Manager**

Компонента є відповідальною за створення списку покупок, наповнення його, зміну, чи видалення.

- **Shop Manager**

Компонента функціонально відповідає за облік магазинів роздрібної торгівлі в системі, товарів в них і т.д.

- **Service Locator**

Компонента відповідальна за пошук магазину роздрібної торгівлі на базі GPS координат користувача.

### **7.1.3. Сервісний та презентаційний рівні**

Сервісний рівень забезпечує взаємодію між клієнтом та сервером додатку (application server).

Фактично, сервісний рівень представляє собою набір REST контролерів, що своєю сукупністю визначають API (Application programming interface) – програмний інтерфейс прикладення. Через цей інтерфейс система надає доступ до свого функціоналу в уніфікованій, строго визначеній та задокументованій формі.

Презентаційний рівень, в свою чергу, відповідає візуальне оформлення інформації, що відображається на веб-сторінці у веб-браузері.

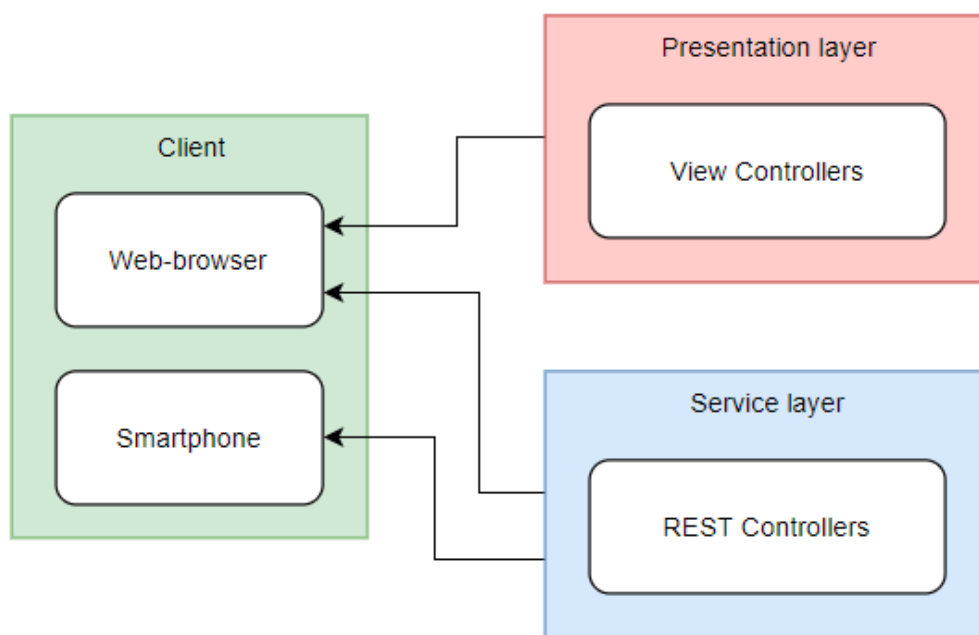


Рисунок ? – Архітектура рівней сервісу та презентації

Презентаційний рівень фактично являє собою сукупність HTML, CSS та Javascript файлів.

Обидва рівні працюють та приймають запити через HTTP протокол. Задля забезпечення безпеки та приватності, вся заємодія, що стосується користувача, як передача логіна/пароля, створення кошику, будь-які операції з кошиком, має проходити через безпечну версію протоколу HTTPS.

#### **7.1.4. Архітектура мобільного прикладення**

Мобільне прикладення у випадку системи самостійного сканування має досить лаконічну архітектуру за рахунок того, що фактично весь об'єм роботи проводиться на серверній стороні. Мобільне прикладення виступає клієнтом системи самостійного сканування. Будь-яка операція, ініційована користувачем, трансформується в зрозумілу серверу прикладень мову та передається на подальшу обробку.

Завданням мобільного прикладення є надання інтуїтивно зрозумілого інтерфейсу користувачу смартфона та забезпечення зв'язку з сервером прикладень (application server).

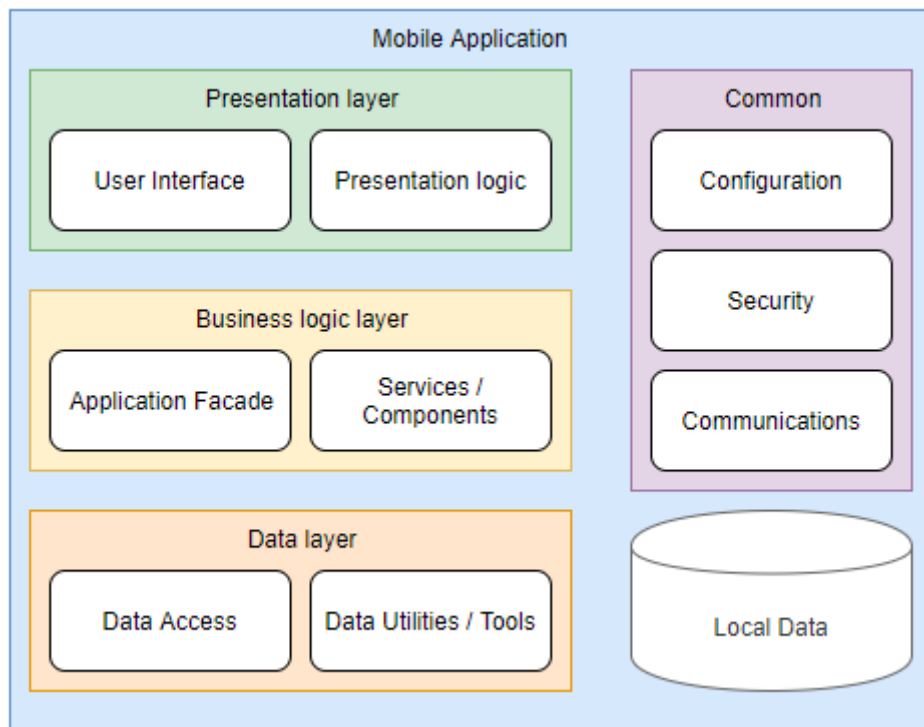


Рисунок ? – Архітектура мобільного прикладення

Розглянемо архітектуру мобільного прикладення на рисунку (?):

### **Presentation layer (Рівень презентації)**

- User Interface

Інтерфейс користувача відповідає повністю за візуальне оформлення прикладення

- Presentation logic

Включає в себе весь програмний код, що забезпечує роботу інтерфейсу користувача – наприклад, перехід з одного екрану на інший, спрацювання відповідних лісенерів, обробка подій натиску на кнопки та інших взаємодій з екраном смартфона в рамках прикладення.

### **Business logic layer (Рівень бізнес-логіки)**

- Application Facade

Являється прошарком між рівнем презентації та бізнес логіки. Надає весь функціонал, який є необхідним для успішного функціонування прикладення. Взаємодіє з компонентами та сервісами, що лежать нижче.

- **Services / Components**

Сервіси та компоненти, які відповідають за певну область функціоналу. У випадку прикладення, що є клієнтом для системи самостійного сканування, за бізнес логіку відповідатиме 2 сервіси. Перший займатиметься інтеграцією з бібліотекою Google Vision для сканування QR коду, інший реалізуватиме виклики на серверну сторону у відповідь на дії користувача в системі.

### **Business logic layer (Рівень бізнес-логіки)**

- **Data Access**

Реалізація доступу до локального сховища

- **Data Utilities / Tools**

Реалізація необхідних для доступу до бази даних інструментів

### **Common (Загальний рівень)**

- **Configuration**

Підтримка конфігурації та кастомізації прикладення. Конфігурація може бути самою різноманітною, наприклад, користувач може мати можливість задати на рівні мобільного прикладення свій улюблений магазин роздрібної торгівлі. В цьому разі не буде виникати необхідності в передачі координат користувача при створення кошику, система відразу знатиме ідентифікатор магазину.

- **Security**

Забезпечення безпеки з'єднання, приватності даних, відсутність витоків даних, нечутливість прикладення до різноманітних вірусів чи інших загроз.

- Communications

Забезпечення всіх необхідних комунікацій. В контексті системи самостійного сканування – забезпечення з'єднання з сервером прикладень через HTTPS протокол.

## 7.2. Діаграми послідовності

Діаграми послідовності відображають взаємодію між об'єктами системи з упорядкуванням за часом. Це дозволяє ясно та прозоро розуміти роль кожного з об'єктів в роботі системи та логічний ланцюжок дій, які виконуються всередині системи.

Розглянемо основні діаграми послідовності для системи самостійного сканування.

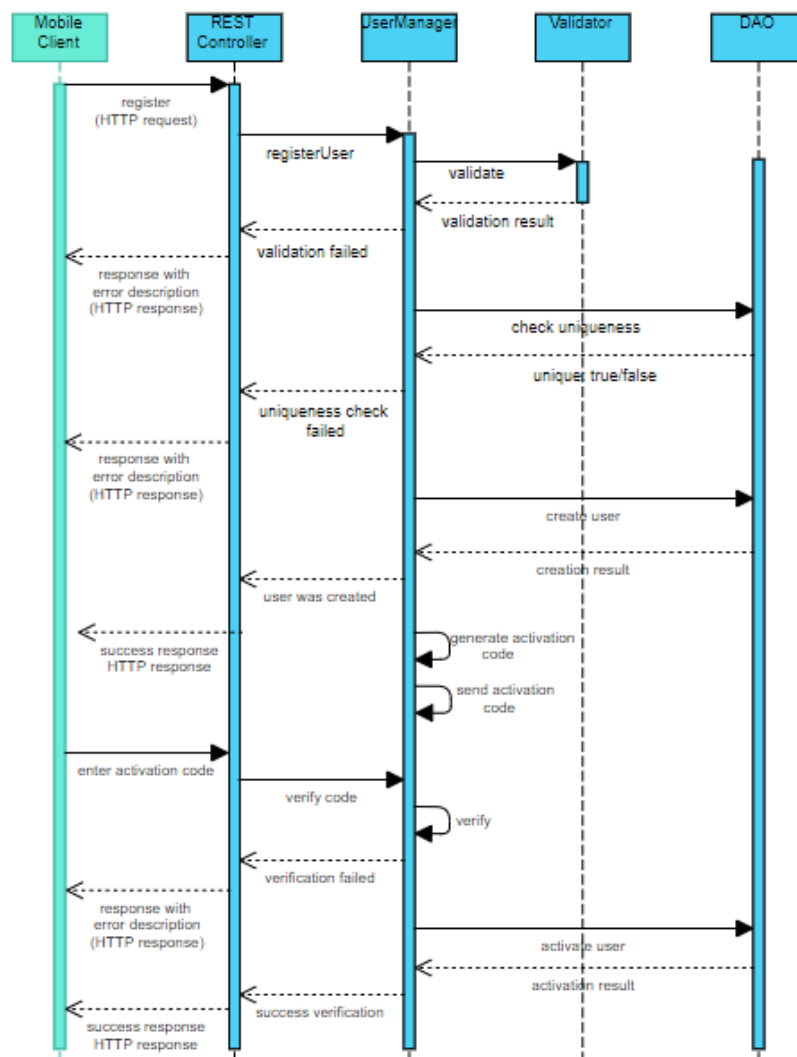


Рисунок ? – Діаграма послідовності для процесу реєстрації в системі

Реєстрація користувача (Рисунок ?):

- Після заповнення форми реєстрації, дані користувача відсилаються на сервер
- RESTController приймає запит, ідентифікує його як запит на реєстрацію та викликає UserManager
- UserManager спершу, за допомогою Validator, валідує запит на правильність введених даних. Якщо дані невірні, то через RESTController формується відповідь з описом помилки та надсилається назад до клієнта
- В разі успішної валідації, UserManager перевіряє унікальність введених даних. Для цього робиться запит до бази даних на предмет перевірки чи вже існує такий користувач з такими даними в системі.
- Якщо дані не проходять перевірку на унікальність, через RESTController формується відповідь з описом помилки та надсилається назад до клієнта.
- В разі успішної валідації, UserManager створює користувача в системі та RESTController надсилає відповідь з успішним статусом.
- Після цього, UserManager генерує код активації та надсилає його на обрану користувачем опцію (номер телефону або електронну пошту)
- Користувач, отримавши код активації, має ввести його в мобальному прикладенні, в результаті чого буде створено запит до сервера на предмет активації користувача.
- UserManager верифікує код активації. Якщо верифікація пройшла неуспішно, RESTController надсилає відповідь з детальним описом помилки назад до клієнту.
- В разі успішної верифікації, UserManager активує створеного в системі користувача, а RESTController відсилає відповідь з успішним статусом.

Процес реєстрації, як і всі інші процеси, що проходять через RESTController є однаковим для будь-яких клієнтів, що підключаються до сервера та взаємодіють з web-сервісом, не залежно від того клієнтом є мобільне прикладення чи веб-браузер.

Наступним процесом розглянемо сканування товарної одиниці та додавання її до кошику.

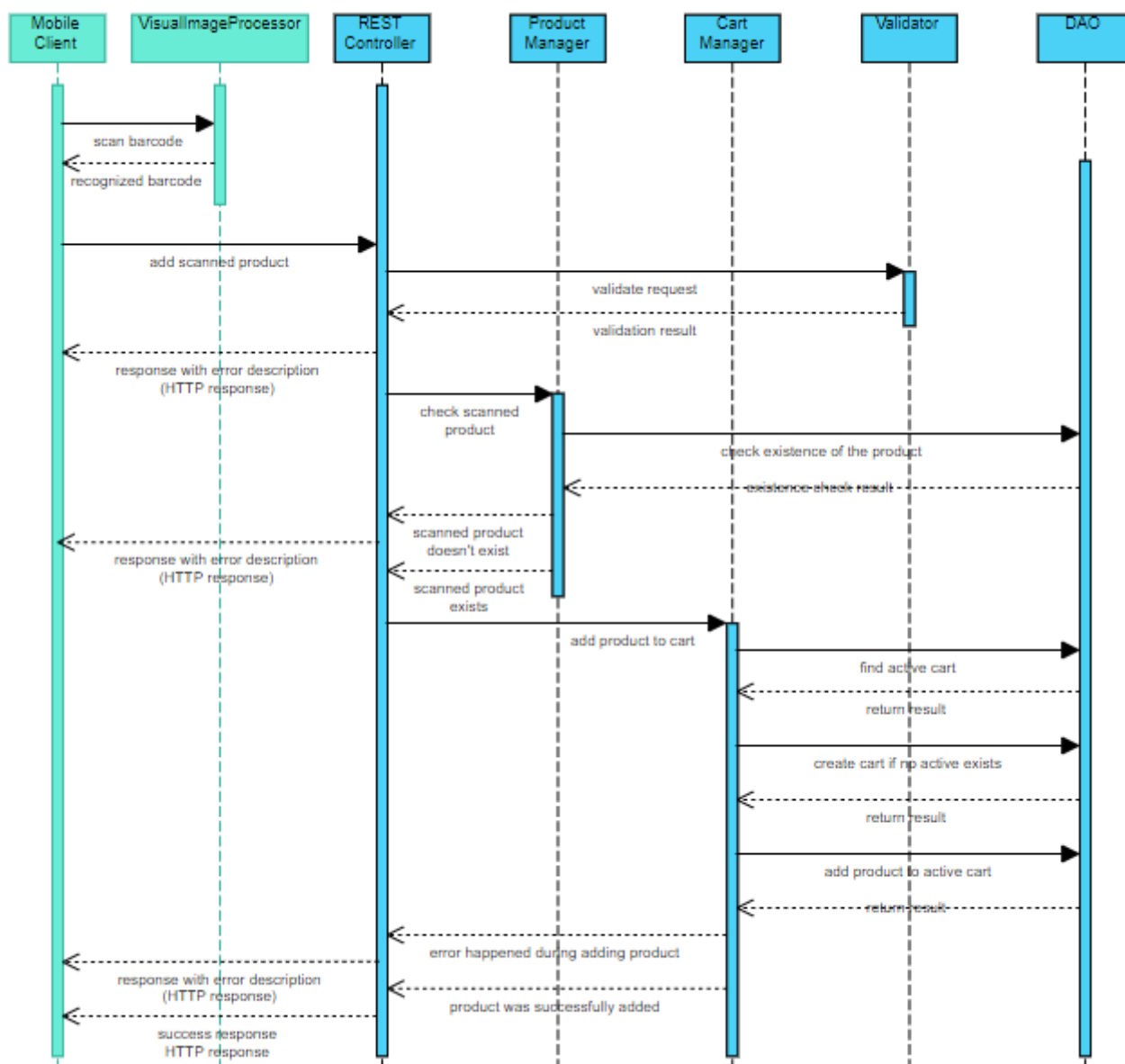


Рисунок ? – Діаграма послідовності для процесу сканування товарної одиниці та додання її до кошику

Сканування товарної одиниці та додання її до кошику (Рисунок ?)

- Користувач за допомогою смартфону сканує штрих- чи QR-код.



- VisualImageProcessor розпізнає цей код та визначає ідентифікатор просканованої одиниці товару
- Користувач додає просканований товар до кошика. Це породжує запит до сервера на додавання товару.
- RESTController приймає запит та валідує його через Validator.
- В разі невалідності запиту, відповідна відповідь з описом помилки надсилається назад до користувача
- Після успішної валідації, відбувається звернення до ProductManager з ціллю перевірки існування такого продукту в системі.
- Якщо ProductManager не знайшов такий товар в системі, формується відповідна відповідь та надсилається назад до клієнта
- В разі успішної перевірки товарної одиниці наявності в системі, викликається CartManager для додавання продукту в кошик.
- CartManager перевіряє, чи існує у користувача активний кошик. Якщо активного кошику немає, то такий створюється.
- Cart Manager додає товарну одиницю в кошик, роблячи необхідні зміни в базі даних.
- Якщо операція проходить успішно, користувачу надсилається відповідь з успішним статусом та всією необхідною інформацією для відображення кошика в мобільному додатку.
- В разі неуспішності, клієнту надсилається відповідний опис помилки.

### 7.3. Протоколи HTTP та HTTPS

HTTP (HyperText Transfer Protocol) протокол відноситься до прикладного рівня моделі OSI та TCP/IP стеку. Початково протокол розроблявся та використовувався для передачі гіпертекстових документів, проте на сьогодні цей протокол широко використовується для будь-яких клієнт-серверних взаємодій.

Протокол передбачає наявність двох учасників:

- Споживач

Споживачом являється клієнт, тобто програмне забезпечення, котре ініціює з'єднання, надсилає через нього запити та приймає і оброблює відповідь

- Постачальник

Постачальником послуг є сервер, який очікує на встановлення з'єднання, отримує запит, обробляє його та відсилає результат обробки назад до клієнта.

Для визначення набору дій, що мають бути виконані, в HTTP проколі використовується URI (Uniform Resource Identifier), який вказується на стороні клієнта. На стороні серверу кожний URI відповідає окремим методам інтерфейсу, який реалізує web-сервіс.

Також, протокол дозволяє вказувати в запиті та відповіді спосіб представлення даних, їх формат, кодування та інші параметри. Подібна прозорість значно спрощує комунікацію між клієнтом та сервером – перший знає формат відправлення та формат відповіді, сервер розуміє надіслане йому повідомлення та формує зрозумілу для клієнту відповідь.

Кожен запит/відповідь складається з трьох частин:

- стартовий рядок;
- заголовки;
- тіло повідомлення, що містить дані запиту, запитаний ресурс або опис проблеми, якщо запит не виконано.

Стартовий рядок має наступний формат:

<Метод> <URI> HTTP/<Версія>

Метод HTTP (англ. HTTP Method) - послідовність з будь-яких символів, крім керуючих і роздільників, яка вказує на основну операцію над ресурсом. Зазвичай метод являє собою короткий англійське слово, записане великими літерами.

Сервер може використовувати будь-які методи, не існує обов'язкових методів для сервера або клієнта. Якщо сервер не розпізнав вказаний клієнтом метод, то він повинен повернути статус 501 (Not Implemented). Якщо серверу метод відомий, але він непридатний до конкретного ресурсу, то повертається повідомлення з кодом 405 (Method Not Allowed). В обох випадках сервера слід включити в повідомлення відповіді заголовок Allow зі списком підтримуваних методів.

Перелік методів, що підтримуються прикладенням:

- GET

Використовується для надання інформації про запитаний ресурс. Всі параметри, які є необхідними для ідентифікації необхідного ресурсу, вказуються в URI запити. Запит з методом GET не містить в собі тіла повідомлення.

- POST

Використовується для обробки даних, переданих користувачем. Дані передаються в цьому випадку в тілі повідомлення.

- DELETE

Використовується для видалення вказаного ресурса. Також не містить в собі тіла повідомлення.

HTTPS (HyperText Transfer Protocol Secure) протокол є розширенням протоколу HTTP, яке було створене з ціллю підвищення безпеки з'єднання, а відповідно і персональних даних. Дані в протоколі HTTPS передаються через криптографічні протоколи TLS та SSL. Протокол SSL вважається устарівшим з 2015 року. TCP-порт, що використовується протоколом HTTPS за замовченням є порт 443.

Протокол забезпечує захист від атак, в якії зайдіяне прослуховування TCP трафіку. Проте, такий захист можливий лише за наступних умов:

- Використання шифрування
- Наявність перевіреного сертифікату на сервері
- Наявність довіри до сертифікату на сервері

Використання HTTPS протоколу починається з установки сертифікату відкритого та закритого ключів для конкретного веб-сервера. В TLS використовується асиметрична схема шифрування, так і симетрична.

Асиметрична схема шифрування дозволяє за допомогою відкритого криптографічного ключа створити загальний секретний ключ.

Симетрична схема шифрування використовує не відкритий ключ, а секретний, та приймає участь в шифруванні ним обміну даних.

Сертифікат веб-серверу фактично є підтвердженням власності на відкритий ключ. Під час встановлення з'єднання, сертифікат та відкритий ключ надсилаються на сторону клієнту, який, отримавши їх, приймає рішення щодо довіри чи недовіри до даного ресурсу. Приватний ключ застосовується задля розшифрування запитів, надісланих клієнтом.

#### **7.4. Інтеграція з Google ML (Machine Learning) Kit**

Google ML Kit – це пакет API функцій, які надають доступ до експертизи компанії гугл в сфері Machine Learning.

Одним з критеріїв вибору цього пакету була його інноваційність та доступність основних моделей нейронних мереж, а також набір інструментів для розпізнавання зображень, різноманітних типів баркодів та навіть облич.

Цей пакет бібліотек дозволяє також будувати свої структури та використовувати їх за умови невідповідності наявних інструментів поставленим вимогам. Це особливо актуально в світлі подальшого розвитку системи самостійного сканування та дозволить в майбутньому розширити функціонал системи задля забезпечення безпеки та охорони.

Інтеграція з API дуже проста, визначається клас, який є процесором для обробки зображень з камери, наприклад VisionImageProcessor.

Основним методом, який необхідно реалізувати є метод process, який займається обробкою вхідного зображення.

```
public void process(Bitmap bitmap, final GraphicOverlay
```

```
        graphicOverlay) {  
        ...  
    }
```

Тут Bitmap – є представленням отриманого зображення з камери.

Далі, з його допомогою необхідно отримати FirebaseVisionImage використовуючи наступний метод:

```
FirebaseVisionImage.fromBitmap(bitmap)
```

Задля визначення баркоду, необхідно створити змінну детектор:

```
FirebaseVisionBarcodeDetector detector = FirebaseVision.getInstance().getVisionBarcodeDetector();
```

У детектора можна викликати код, що поверне завдання зі списком розпізнаних баркодів:

```
Task<List<FirebaseVisionBarcode>> task = detector.detectInImage(image)
```

Після успішного виконання цього завдання можна подальшим чином виконувати обробку змісту, що був закладений у баркоді:

```
FirebaseVisionBarcode.getRawValue()
```

Таким чином, мобільне прикладення може за допомогою камери сканувати штрихкоди та QR-коди товарів, розпізнавати їх та відсилати інформацію на сторону серверу з ціллю запита інформації по просканованому товару або додавання його до кошику.

## 8. РОЗРОБЛЕННЯ ІНТЕРФЕЙСА КОРИСТУВАЧА

### 8.1.Інтерфейс користувача

Вікно входу(рис?) має стандартний набір функцій та передбачає можливість реєстрації в системі. Логін вводиться в звичайне текстове поле, а поле паролю містить функціонал, що маскує введені дані символом крапки. Альтернативою класичному входу є функція безпарольового доступу за допомогою відбитка пальця, що буде доступною на пристроях, які мають сканер.Присутня функція відновлення паролю.

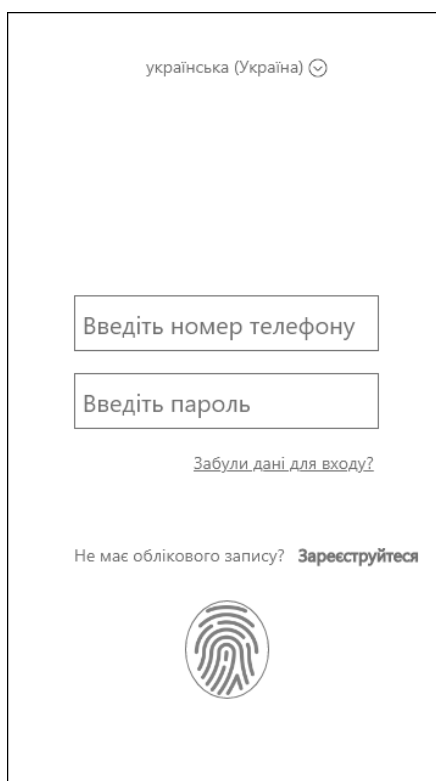


Рисунок ? – Вікно входу

Присутня функція відновлення паролю, що направляє користувача на нове вікно, де пропонується ввести в текстове поле електронну адресу або номер телефону для відновлення доступу до облікового запису. Далі процес відновлення паролю представляється користувачеві у вигляді послідовних вікон, що мають текстові поля для взаємодії: ведення коду активації, вікна введення нового паролю та інформаційного вікна, що перенаправить користувача на вікно входу. Загальний прототип інтерфейсу користувача вказаний у Додатку 7.

Вікно реєстрації (рис ?) містить стандартний функціонал для взаємодії. Подальший процес реєстрації представить користувачеві два послідовних вікна: вікно активації та інформаційне вікно, що перенаправить користувача на вікно входу.

Регістрація

Ім'я

Прізвище

Номер телефону

E-mail

чч.мм.рррр

Адреса

Далі

Рисунок

Головна сторінка (рис ?) ділиться на три зони. Верхня зона відображає потрібні для користувача дані, пов'язані з системою лояльності магазину. Центральна зона інформативна, де відображається акції або інформація, яку магазин вважає корисною для своїх клієнтів. Нижню зону головного екрану займає панель вкладок: «Головна», «Пошук товару», «Сканування», «Історія» та «Меню».

Персональні пропозиції

Бали

Копійковий рахунок

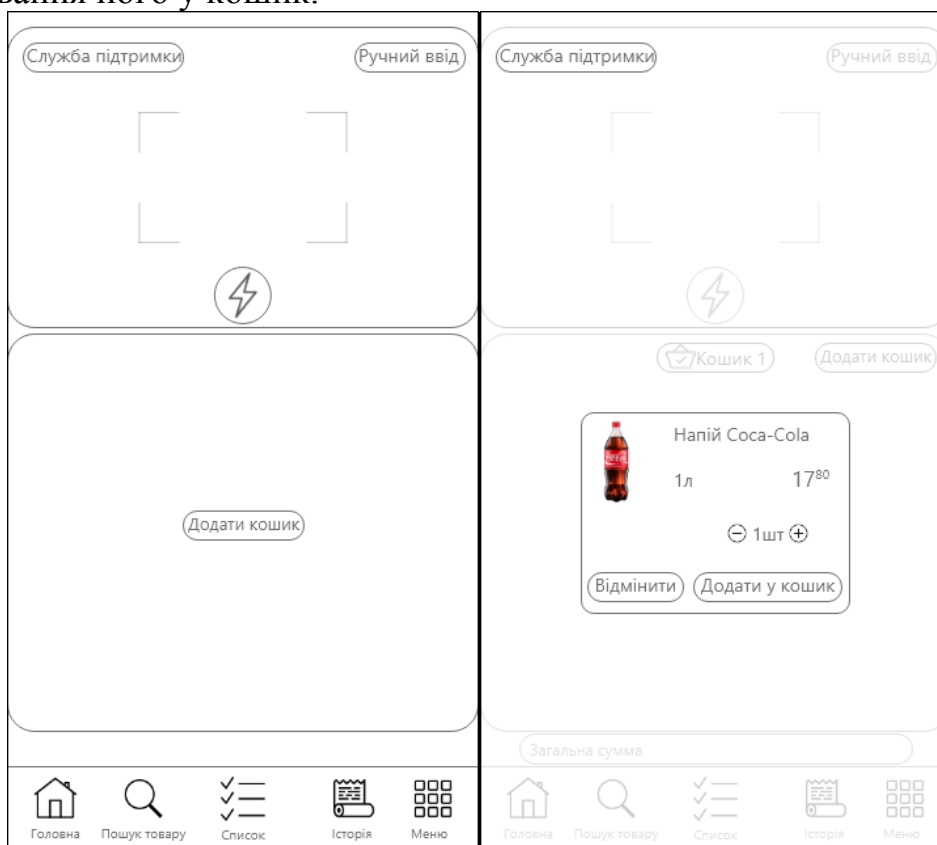
//Акції або новини

Головна Пошук товару Сканування Історія Меню

Рисунок

Вкладка «Пошук товару» має типовий набір функцій, що передбачений специфікою даного процесу. У верхній частині вікна відображається поле пошуку та спливаюче поле категорій, до яких відноситься шуканий товар. А в центральній частині відображається список товарів які характеризуються ключем пошуку.

Вкладка сканування товару(рис ?) відкривається відповідним пунктом «Сканування» на панелі вкладок головного екрану. У верхній частині екрану знаходиться зона для сканування штрих-кодів, що має додатковий функціонал для активації спалаху камери, зв'язок із службою підтримки та ручний ввід коду товару. Центральну частину замає інтерфейс функціоналу, пов'язаний із взаємодією з кошиком. Коли користувач сканує товар, в центральній частині спливає вікно з описанням товару, де відображено відповідний функціонал, що забезпечує вибір кількості бажаного товару та додавання його у кошик.

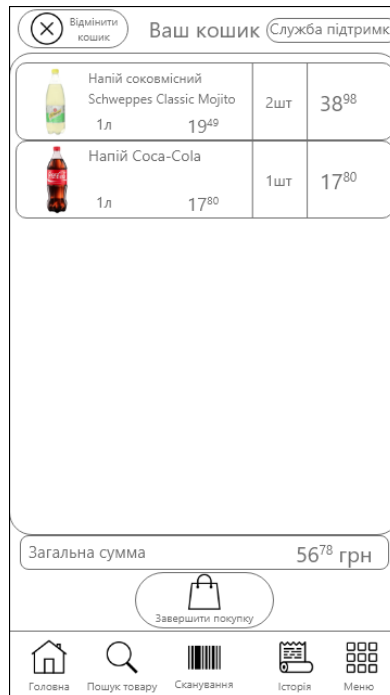


Рисунок

Рисунок

Редагування кошика здійснюється у вкладці «Список»(рис), та передбачає зручне редагування вмісту кошика. Натискаючи подвійним дотиком на пункт з товарною одиницею, користувач має можливість відредагувати кількість даного товару. Видалення товарної одиниці зі списку кошика відбувається горизонтальним зсувом бажаного пункту. Для повного анулювання кошика передбачений пункт «відмінити кошик», що викликає спливаюче вікно , в якому користувач має дати згоду на продовження даної дії.



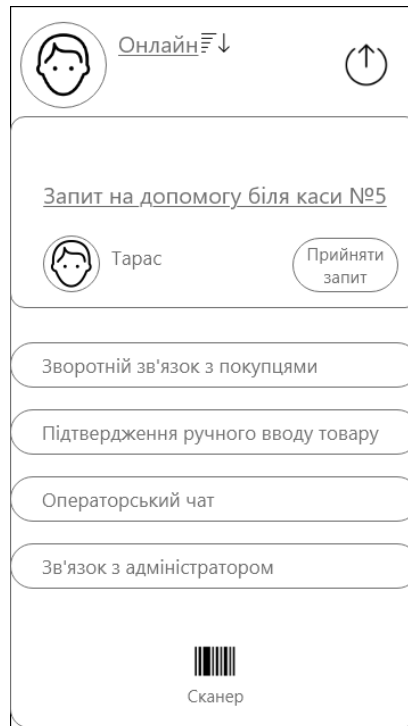


Рисунок

Для завершення процесу покупки користувач має вибрати пункт «Завершити покупку»

## 8.2. Інтерфейс оператора

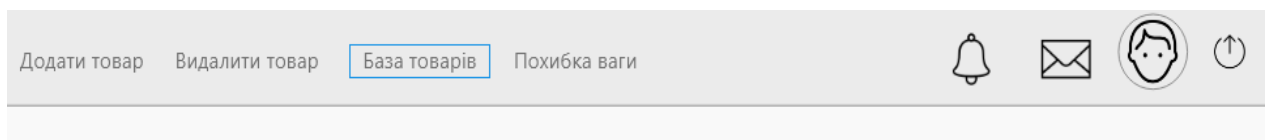
Вікно входу для оператора має стандартний набір функцій, що відображені пунктом «Увійти в систему» та двома текстовими полями, для введення логіну та паролю. Головний екран оператора (рис) складається з пунктів «Зворотній зв'язок з покупцем», «Підтвердження ручного вводу товару», «Операторський чат», «Зв'язок з адміністратором» та інформаційного поля, що призначено для системних повідомлень та пункту статусу. Пункт «Зворотній зв'язок з покупцями» (додаток 8) являє собою типовий список діалогів з покупцями з можливістю зручної навігації. Коли оператор знаходиться на інших вікнах застосунку, при виникненні системних повідомлень у верхній частині екрану буде спливати інформаційне вікно з вказівками для оператора. Пункт статус містить в собі підпункти «онлайн», «оффлайн», «обід», «особиста перерва».



Рисунок

### 8.3.Інтерфейс адміністратора

Вікно входу для адміністратора має стандартний функціонал. Меню додатка складається з позицій «Додати товар», «Видалити товар», «База товарів», «Похибка ваги», «Сповіщення», «Повідомлення», «Вихід». Пункт «Похибка ваги» містить список магазинів мережі, вибравши один, адміністратор має змогу внести правки в налаштування ваг.



рисунок

## 9 СТАРТАП-ПРОЕКТ

### 9.1 Загальний стан ринкового середовища для впровадження проекту

Роздрібна торгівля сьогодні має справу з сотнями тисяч товарів, які продавець може можливість запропонувати покупцям, для чого йому потрібно здійснювати великі капітальні витрати на побудову і оснащення торгових приміщень, а також придумувати нові підходи до інформування покупців про наявність звичних товарів і появу нових, з новими властивостями.

З іншого боку товарного прилавка, при сучасному розмаїтті товарів та послуг покупець стає заручником величезних торгівельних площ, серед яких йому приходится витрачати багато часу на те, щоб знайти і придбати потрібну річ.

Таким чином у сфері роздрібної торгівлі обидві сторони і продавці і покупці зацікавлені в використанні інструментів, які продавцям полегшують оперування величезними товарними масами і донесення їхньої пропозиції потенційним покупцям, а самим покупцям полегшують вибір товару і знижують витрати часу на його придбання.

Таблиця 10.1 ілюструє сучасну тенденцію розвитку мереж роздрібної торгівлі, яка характеризується тим, що в побудові ринкової стратегії ритейлери опираються на доступні можливості інформаційної епохи[ 1 ] [ 2 ]. З кожним роком поглиблюється «диджиталізація» їхнього бізнесу, що дозволяє підвищити його ефективність і досягнути кращого співвідношення «прибуток/витрати». Саме цим обумовлена поява кас самообслуговування та мобільних додатків.

Таблиця 10.1

№ п/п	Назва	Кільк. магазинів	Формат	Нові формати	Self Checkout	Мобільний додаток	Кількість завантажень мобільного додатку
1	АТБ	882	дискаунтер	Експрес		+	> 1 млн.
2	Фора	239	магазин біля дому		+	+	
3	Сільпо	236	супер/гіпер		+	+	> 100 тис.
4	Наш Край	225	супер/магазин біля дому/ експрес			+	
5	ЕКО маркет	114	супер/гіпер/ магазин біля дому				
6	Таврія В	76	гіпер/ магазин біля дому	Інтернет-магазин	+		
7	Varus	59	супермаркет / магазин біля дому			+	> 100 тис.
8	Novus	44	супер/гіпер/ магазин біля дому		+		
9	Billa	33	супермаркет				
10	Metro C&C Ukraine	26	центр опт. торг. (cash&carry)	Бери вези		+	> 100 тис.
11	Велика Кишеня	24	супер/гіпер			+	> 50 тис.
12	Велмарт	15	супер/гіпер		+	+	> 10 тис.
13	Ашан	11	гіпермаркет-дискаунтер	Мій Ашан	+	+	> 100 тис.
14	FOZZY	10	центр опт. торг. (cash&carry)				

Ідея залучення і утримання покупців з використанням сучасних гаджетів спонукає ритейлерів вкладувати кошти в розробку і підтримку додатків, які можна завантажити в смартфон з ресурсів Apple Store та Gogle Play. Станом на сьогодні [ ] додатки такого призначення вже пропонують основні гравці на ринку роздрібної торгівлі (наприклад, АТБ, Фора, Сільпо, Metro C&C, Ашан, Велика кишеня. та інші). Поряд з цим, деякі відомі ритейлери ще не оцінили всіх можливостей, що відкриваються з приходом цифрової епохи, наприклад, Novus, Billa, Фуршет, ЕКО маркет.

Індивідуальні гаджети (смартфони, планшети), які стали елементом повсякденного побуту і до яких користувачі звертаються щодня з різних питань сотні разів, традиційно використовуються як вікно у світ інформації. Саме ця можливість донесення до покупця рекламної інформації та інформації про акції використовується ритейлерами в першу чергу. Але сучасні гаджети забезпечують значно ширші можливості для формування у їхніх власників бажання прийти до магазину та купити товар.

Для успішної продажі покупець і продавець повинні дійти згоди, щодо якості та ціни товару. З точки зору покупця товар має бути високої якості та мати справедливую ціну (припускаємо, що функціональність товару відповідає вимогам покупця). Очевидно, що обидва названі критерії вибору, які мають для покупця визначальне

значення при прийнятті рішення про покупку, носять суб'єктивний характер, а значить, піддаються маніпулятивним технологіям, до яких часто звертаються недобросовісні продавці через рекламу та акції. Тому головною запорукою стабільного успіху для ритейлера є лояльність покупців, яка базується на спогадах про вдачу покупку. Саме тому, щоб спогади не вгасали, ритейлер повинен їх регулярно підживлювати нагадуваннями про місце, з яким вони пов'язані і одночасно інтригувати покупця новими пропозиціями. Сучасний гаджет, який завжди поряд зі своїм хазяїном, і напоготові, забезпечує найкоротший інформаційний канал для подачі потенційному покупцю цікавої та корисної профільної інформації, індивідуально підготовленої на основі дослідження його (покупця) уподобань..

## 9.2 Опис ідеї проекту

Ідея проекту - створити систему підтримки роботи закладів роздрібної торгівлі, яка оснований на найбільш ефективному способі самообслуговування - самостійного сканування покупцями товарів за допомогою власних гаджетів.

У результаті реалізації проекту і покупці і продавці отримують вигоду, яка, одного боку, сприятиме росту лояльності покупців до торгівельної марки, а з другого боку, підвищуватиме якість обслуговування і прибутки продавця.

В таблиці 10.2 перелічені переваги від реалізації проекту, які отримують покупці.

Таблиця 10.2

№№ п/п	Можливості системи	Вигоди для користувача
1	Повне самообслуговування при здійсненні покупок, що включає: – попереднє складання списку покупок;	Зручність користування продуманим інтерфейсом на знайомому пристрої, економія часу при виборі товару та при виході з торгового залу з мінімальною затримкою на оплату

	<ul style="list-style-type: none"> <li>– навігація по торговому залу по оптимальному маршруту для заповнення кошика товарами із складеного списку;</li> <li>– самостійне сканування товарів смартфоном перед тим, як покласти в кошик;</li> <li>– відображення вмісту кошика на екрані у вигляді списку товарів з ціною і загальною сумою</li> <li>– редагування списку товарів у кошику (зміна кількості одиниць, видалення позицій)</li> <li>– самостійний розрахунок за товар після закінчення заповнення кошика на касі без участі персоналу магазину.</li> </ul>	
2	Планування покупок	Мінімізація витрат часу на відвідування магазину.
3	Інформація про наявність товарів із сформованого списку та про акційні пропозиції	Зручність при плануванні покупок

Переваги, які отримує ритейлер при впровадженні проекту, ілюструє таблиця 10.3.

Таблиця 10.3

№ п/п	Можливості системи	Вигоди для ритейлера
	Персональна взаємодія покупця з торговою системою	Ріст лояльності покупців до торгової марки за рахунок створення покупцю зручного середовища для здійснення покупок.  Ріст ефективності витрат на рекламу за рахунок її індивідуального характеру, в основі якого лежить аналіз смаків та потреб конкретного покупця, а не віртуальної фокус-групи.
	Повне самообслуговування покупців	Заощадження коштів та площі торгівельного залу на організацію касових вузлів.
	Скорочення витрат часу покупця на придбання набору товарів	Збільшення прибутків за рахунок росту ефективності використання ресурсів магазину.

### 9.3 Порівняння проекту з існуючими аналогами

В таблиці 10.1 наведені дані про використання торгівельними мережами у взаєминах покупець-продавець нових інструментів: кас самообслуговування та мобільних додатків. Варто відзначити, що серед всіх наведених в таблиці 10.1 засобів лише торгова марка «ТАВРИЯ В» впровадила технологію кас самообслуговування з самостійним скануванням товарів засобами гаджетів покупців. Решта торгових мереж впровадили каси самообслуговування, які не співпрацюють з мобільними додатками

в межах ланцюжка: «товар - каса - оплата». Ці ритейлери використовують гаджети як допоміжні засоби інформаційної підтримки процесу продажу товарів, а саме, для передачі покупцю рекламної інформації, а режим сканування використовується як довідковий для надання покупцеві інформації про вартість конкретного зразка товару. В таблиці 10.4 наведені порівняльні характеристики запропонованої системи та існуючих.

Таблиця 10.4

№ п/п	Характеристика	Назва проекту			
		SMARTSHOP	АТБ	ФОРА	Таврія В
1	Попередня підготовка списку покупок	+	+	-	-
2	On-line навігація по торговельному залу	+	-	-	-
3	Сканування товару	+	-	+	+
4	Редагування вмісту кошика	+	+	-	+/-
5	Передача списку покупок на касу	+	-	-	+

Коротка характеристика відмінностей запропонованого проекту від аналогів (a), (b), (c):

1. Попередня підготовка списку бажаних товарів використовується в конкурентних продуктах як пам'ятка. У запропонованому проекті список окрім пам'ятки виконує функцію завдання для побудови оптимального маршруту по торговельному залу для послідовного підбору продуктів та заповнення кошика.
2. On-line навігація по торговельному залу підказує покупцю найкоротший шлях від поточного його положення до місця розташування чергового товару із списку покупок.
3. Сканування і ідентифікація товару відбуваються аналогічним чином в запропоновано-му проекті та в додатку «ТАВРИЯ В» для занесення товару в кошик. У додатку «ФОРА» сканування служить лише для інформування покупця про ціну товару.



4. При заповненні кошика товарами не рідко виникає потреба «вийняти» товар з кошика і вернути на полицю. У додатку «Таврія В» ця операція реалізована як видалення рядка у списку товарів. У запропонованому проекті видалення товару з кошика можливе аналогічним чином - шляхом пошуку відповідного рядка у списку покупок та його видалення, а можливе шляхом ідентифікації зайвого товару шляхом його повторного сканування у режимі видалення.
5. У запропонованому проекті реалізована додаткова можливість при роботі з касовим вузлом - розділити список на кілька частин і сплатити за кожну з них окремо та отримати за кожну оплату окремий чек.

#### **9.4 Технологічний аудит проекту**

Оскільки метою проекту є розробка системи з чітко прописаними вимогами і чітко окресленим середовищем, в якому вона повинна функціонувати, то для



втілення проекту у життя оптимальною є каскадна модель життєвого циклу проекту

[ 3 ], Рис. 10.1

наведена на рис. 10.1.

Вона характеризується тим, що:

- всі стадії проекту виконуються строго послідовно, що дає можливість чітко планувати строки виконання етапів та відповідні ресурси (грошові та людські);
- вимоги лишаються незмінними на протязі всієї розробки, що скорочує час виконання проекту

На сьогоднішній день створення системи знаходиться на стадії розробки моделі прототипу. Наступним кроком на шляху впровадження проекту буде демонстрація моделі прототипу і пошук замовників, які забезпечать фінансування подальшої роботи над системою.

Всі технології, необхідні для реалізації проекту, існують і доступні на ринку програмних продуктів. Зокрема, вимогами до системи, що розробляється, найкраще відповідає клієнт-серверна модель. При цьому, реалізація функціоналу системи розподіляється між мобільним клієнтом у смартфоні покупця і сервером, на який покладається основне навантаження по взаємодії з робочими базами даних.

Оскільки архітектура системи має розподілений характер, то питання стабільності роботи та захисту інформації буде вирішуватись з залученням *REST* веб сервісів.

Розробка прототипу і подальша робота над *frontend* додатком клієнта ведеться на основі *Web-based інтерфейсу* на платформі *Android*, як найбільш поширеній для мобільних пристроїв.

Для розробки інтерфейсу користувача та організації взаємодії мобільного клієнта з сервером баз даних будуть використані інструменти *HTML/CSS/JavaScript, JSP, Spring фреймворк*.

## 9.5 Аналіз ринкових можливостей запуску стартап-проекту

### і стратегія виходу на ринок

За результатами 2018 року перші місця за обсягами виручки та прибутків серед ритейлерів, що займаються продажем продуктів харчування розподілилися так як показано в таблиці 10.5 [ 4 ]:

Крім представлених у таблиці 10.5 є ще ряд компаній з першої двадцятки лідерів ритейлу, наприклад Ашан, Vagus, Велика кишеня, Фора, які активно розвивають свої мережі і впроваджують нові форми організації торгівлі, але у жодній з них не реалізована сама прогресивна технологія самообслуговування покупців з самостійним скануванням.

Таблиця 10.5

Компанія		Виручка-2018, млрд грн	Прибуток-2018, млн грн
АТБ		85,73	2723
Metro Cash&Carry Ukraine		17,42	467
Novus		9,1	290

Додатки, які використовуються у магазинах перелічених мереж виконують лише функцію інформаційної підтримки покупців, що включає, наприклад, надання інформації про розташування та режим роботи магазинів, інформації про акції, деякі

дозволяють сканувати код товару і отримувати інформацію про його ціну. Жодна із згаданих торгових мереж не має реалізованої системи самообслуговування з самостійним скануванням. Тому саме на ці компанії будуть спрямовані маркетингові зусилля для зацікавлення у впровадженні запропонованої нами системи.

З огляду на специфіку ринкового середовища, де існує потреба в результаті розробки стартап-проекту оптимальними стратегіями будуть:

- базовою стратегією розвитку після виводу на ринок стартап-проекту має бути спеціалізація;
- базовою стратегією конкурентної поведінки на ринку має бути зайняття конкурентної ніші;
- базовою стратегією позиціонування на ринку має бути: «якість-надійність-оперативність».

## **9.6 Розробка маркетингової програми.**

### **9.6.1 Формування маркетингової концепції товару**

Так як запропонована система реалізації самообслуговування покупців магазинів роздрібної торгівлі є інструментом, який дає ритейлерам змогу підвищити рівень обслуговування клієнтів і тим самим збільшити рівень лояльності загалом покупців до їхньої торгової мережі, то потенційно зацікавленою стороною для інвестування в стартап-проект є саме ритейлери. Тому саме великі компанії роздрібної торгівлі складають фокус-групу, на яку повинні орієнтуватись маркетингові зусилля по впровадженню стартап-проекту.

Ключові переваги, які отримує користувач від впровадження стартап-проекту розкриті в таблиці 10.6.

Таблиця 10.6

<i>№ п/п</i>	<i>Потреба</i>	<i>Вигода, яку пропонує товар</i>	<i>Ключові переваги перед конкурентами</i>
1	Вдосконалення роботи торговельних закладів шляхом	Впровадження технології обслуговування з	1) Запропонована в стартап-проекті система самообслуговування покупців з самостійним

	впровадження інноваційних технічних рішень, які підвищують якість обслуговування і стимулюють зростання лояльності покупців.	самостійним скануванням товарів покупцями з мінімальними капітальними вкладеннями за рахунок використання при здійсненні покупок їхніх власних гаджетів .	скануванням товарів характеризується меншими капітальними витратами за рахунок використання покупцями для сканування товарів власних гаджетів. 2) Запропонована технологія самообслуговування скорочує витрати часу покупців на відвідування магазину.
--	--	---	---

### 9.6.2 Конкурентна модель товару

Конкурента перевага визначається споживчою цінністю товару [5], яка має кілька рівнів ієрархії [ ], що розкриті в таблиці 10.7

Таблиця 10.7

1. Цінність товару за задумом - здатність задовольнити головні потреби	Забезпечує підвищення якості обслуговування та стимулює зростання лояльності покупців при мінімальних капіталовкладеннях
2. Цінність товару у реальному виконанні, що забезпечують втілені в товарі технічні рішення і які визначають його надійність, зручність при використанні, його дизайн та стиль	<p>Покупцеві забезпечується:</p> <ul style="list-style-type: none"> <li>– реєстрація в системі та вхід за допомогою свого логіна та пароля;</li> <li>– можливість створити попередній список товарів для закупки;</li> <li>– підказка, щодо розташування товару у торговому залі;</li> <li>– сканування одиниць товару і отримання актуальної ціни одиниці товару;</li> <li>– можливість створити в системі віртуальний кошик, куди він буде класти відібраний для покупки товар</li> </ul>

	<ul style="list-style-type: none"> <li>– можливість змінювати вміст кошика та контролювати загальну вартість зібраних у кошику товарів;</li> <li>– можливість передати список зібраних у кошику товарів на касовий вузол для розрахунку і виконати оплату за товари</li> <li>– можливість отримувати on-line допомогу служби підтримки</li> </ul>
3. Цінність товару завдяки підтримці товару виробником впродовж його життєвого циклу	<p>На етапі розробки забезпечуються:</p> <ul style="list-style-type: none"> <li>– побудова системи згідно вимог замовника.</li> </ul> <p>На етапі впровадження розробки забезпечуються:</p> <ul style="list-style-type: none"> <li>– доопрацювання та адаптація системи до вимог конкретного середовища її функціонування на різних площадках замовника;</li> <li>– навчання персоналу роботі з системою.</li> </ul> <p>Після впровадження розробки забезпечуються:</p> <ul style="list-style-type: none"> <li>– виконання регламентних робіт з технічними засобами системи</li> <li>– розвиток системи згідно побажань замовника.</li> </ul>

З врахуванням того, що результат стартап-проекту може розраховувати на попит лише з боку представників вузького кола крупних ритейлерів, то цінову пропозицію потрібно формувати з оглядом на те, що буде одиничний продаж. За основу для визначення ціни проекту приймемо трудомістськість етапів 3 - 7 життєвого циклу системи (див. рис. 10.1), наведену в таблиці 10.8.

Таблиця 10.8

Етап	Розробка прототипу	Уточнення вимог	Розробка системи	Тестування	Впровадження та технічна підтримка
------	--------------------	-----------------	------------------	------------	------------------------------------

Тру- домісткість (люд./год.)	1600	640	4800	1600	800
------------------------------------	------	-----	------	------	-----

Таблиця 10.9

1	Орієнтовна загальна трудомістськість розробки та впровадження стартап-проекту	9 440 люд./год.
2	Середня вартість години кваліфікованої праці фахівця в галузі	300 грн.
3	Загальні трудовитрати (3) = (1) x (2)	2 832 000 грн.
4	Нарахування на фонд оплати праці (4) = (3) x (22%)	623 040 г рн.
5	Накладні витрати становлять (5) = (3) x (20%)	566 400 грн.
6	Загальна вартість без ПДВ (6) = (3) + (4) + (5)	4 021 440 грн.
7	ПДВ (7) = (6) x (20%)	670 200 грн.
8	<b>Орієнтовна вартість розробки та впровадження стартап-проекта з ПДВ (8) = (6) + (7)</b>	<b>4 021 000 грн.</b>

Розрахунок орієнтовної вартості впровадження стартап-проєту наведений в таблиці 10.9. Отримана в розрахунку сума є відправною точкою для пошуку інвестора.

Для одиночної продажі оптимальною буде прямий продаж, без посередників.

Першим кроком до встановлення маркетингових комунікацій повинна бути домовленість з потенційним замовником про проведення технічного семінару на його площадці для профільних фахівців з доповіддю про характеристики та можливості системи, яка пропонується, та демонстрація макета прототипу *frontend* додатка системи.

## **9.7 Висновки**

1. Тема стартап-проекту своєчасна і створить для ритейлера, який його впровадить реальні конкурентні переваги в боротьбі за долю ринку завдяки росту лояльності покупців і покращення економічних показників процесу роздрібного продажу.
2. Конкуренція для виводу на ринок стартап-проекту слабо виражена у зв'язку з тим, що розвиток цифрових технологій для підтримки бізнесу ритейлерів ведеться силами фахівців, основними задачами яких є супровід існуючих систем інформаційного забезпечення роботи магазинів, що ілюструється впровадженням додатків для клієнтів, які виглядають зробленими по залишковому принципу.
3. З огляду на існуючий рівень підтримки цифровими технологіями роздрібних продаж стартап-проект має хороші шанси викликати зацікавлення у учасників ринку і отримати фінансування з огляду на незначні витрати на впровадження у порівнянні з прибутками ритейлерів (таблиця 10.5).



## ВИСНОВКИ

У роботі запропонована архітектура системи самостійного сканування для мережі магазинів роздрібної торгівлі, в рамках якої функціонал сканування було перенесено в мобільний додаток на смартфоні. Мобільною платформою було обрано Android за рахунок її популярності, а для сканування та розпізнавання штрих-коду товарних одиниць було обрано пакет бібліотек, що поставляються компанією Google – ML Kit.

Виконано основну задачу роботи – вдосконалено існуючу систему самостійного обслуговування для мережі магазинів роздрібної торгівлі. Проаналізовано цілу екосистему, що використовується в роздрібній торгівлі, та розроблено нову систему самостійного сканування в рамках обраної екосистеми.

На даний момент завершено проектування серверної та клієнтської частини системи.

Розроблено детальну архітектуру, описано всі взаємодії між елементами системи, а також досліджено інструменти, що використовуватимуться під час розробки.

Одним із головних надбань даної роботи є те, що розроблена система не лише може суттєво зменшити час на оплату, що затрачається кінцевим користувачем, чи збільшити прибутковість бізнесу, але і надати нові інструменти до таргетованої взаємодії із кінцевим користувачем (споживачем).

На базі поточного проектування, буде розроблено початковий прототип такої системи, а також розширено архітектуру системи задля покращення інтеграції з екосистемою, що використовується наразі в мережах магазинів роздрібної торгівлі. Також планується підтримка не лише однієї, а багатьох подібних мереж однією системою, що приводить до досліджень в області масштабування системи, високої доступності та інтеграції з хмарковими сервісами й повне перенесення обчислень в хмару.

Розроблена в процесі виконання роботи нова система самообслуговування для мережі магазинів роздрібної торгівлі дозволить будувати на її базі аналітичні комплекси для таргетованої взаємодії із кінцевими користувачами та збільшення на її основі обсягів продажу. Напрацювання даної роботи можуть бути застосовані і розвинуті у нових версіях даної системи.

## Перелік використаних джерел

- 1.1 Системы самообслуживания для магазинов  
<https://www.pilot.ru/catalog/sistemy-samoobslyzhivaniya/>
- 1.2 Antonina Kelman. Каси самообслуговування. За і Проти  
[https://datawiz.io/uk/blogs/kassy\\_samoobslyjyvania\\_za\\_i\\_protivua/](https://datawiz.io/uk/blogs/kassy_samoobslyjyvania_za_i_protivua/)
- 1.3 Самостійне сканування як частина Customer experience  
<https://systemgroup.com.ua/uk/samostiynе-skanuvannya-yak-chastyna-customer-experience>
- 10.1 Ритейл и технологии: как современные компании продвигают продукты через инновации.  
<https://www.retail.ru/articles/riteyl-i-tekhnologii-kak-sovremennye-kompanii-prodvigayut-produkty-cherez-innovatsii/?PageSpeed=noscript>
- 10.2 АТБ, Сильпо или Ашан: чье приложение стоит установить на смартфона  
<https://psm7.com/technology/atb-silpo-ili-ashan-che-prilozhenie-stoit-ustanovit-na-smartfon.html>
- 10.3 Бланк, С. Стартап. Настольная книга основателя / С. Бланк, Б. Дорф ; пер. с англ. Т. Гутман, И. Окунькова, Е. Бакушева. – 2-е изд. – Москва : Альпина Паблишер, 2014. – 614 с.
- 10.4 Топ-200: рейтинг рітейлерів України за доходом, прибутком або збитком  
<https://rau.ua/novyni/top-200/>
- 10.5 Ф.Котлер, Маркетинг менеджмент. Экспресс-курс. 2-е изд. СПб.: Питер, 2006. - 464 с.
- 10.6 Ткаченко Т.П. Концепція трьох рівнів товару.  
[http://www.rusnauka.com/7\\_NND\\_2009/Economics/43053.doc.htm](http://www.rusnauka.com/7_NND_2009/Economics/43053.doc.htm)